

**UNIVERSIDAD CARLOS III DE MADRID**

**PROYECTO FIN DE CARRERA  
INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN  
MAYO 2014**



**TÉCNICAS DE ETIQUETADO Y DESAMBIGUACIÓN  
MORFOLÓGICA DEL INGLÉS CON REDUCIDA  
INFORMACIÓN CONTEXTUAL**

Tutores:

Valentín Miguel Moreno Pelayo

Alumno:

Eduardo García Vírseda

## AGRADECIMIENTOS

Con este Proyecto Fin de Carrera termina mi andadura universitaria, que se ha alargado mucho más de lo que quisiera. Ha habido momentos para la alegría y para la pena, pero finalmente, y con esfuerzo, se ha llegado a la meta.

En primer lugar querría expresar mi más sincero agradecimiento a mi tutor, Valentin Miguel Moreno Pelayo, por la confianza que depositó en mí para la realización del trabajo, su paciencia infinita y su siempre buena predisposición para consultas y ayuda.

Por supuesto, mi gratitud a mi familia, y muy especialmente a mis padres, a mi hermano y a mis “chicas vascas”, por su apoyo, comprensión y fuerza en todo momento.

Muchas gracias a los compañeros de carrera que llegaron a ser amigos, por todos los años y experiencias que compartí, comparto y seguiré compartiendo con ellos. Mención especial para Uge (¡ánimo con el doctorado!), que me recomendó y puso en contacto con Valentín, y ahí comenzó a fraguarse esto.

Mi reconocimiento a todos los autores e investigadores consultados para la realización de este Proyecto Fin de Carrera, reflejados en el apartado de Referencias, ya que sin su excelente trabajo, muy probablemente no podría haber hecho el mío.

Y por último, y no por ello menos importante, sino todo lo contrario, a mi novia, Rocío, la persona que más luz ha aportado para que consiguiera llegar al final. Su ayuda, fuerza, dedicación, influencia y cariño, han resultado determinantes en este viaje.

En resumidas cuentas, muchas gracias, por todo, a todos.

# ÍNDICE

ÍNDICE DE FIGURAS.....	9
ÍNDICE DE TABLAS .....	12
RESUMEN .....	13
ESTRUCTURA DEL DOCUMENTO.....	15
1. INTRODUCCIÓN.....	17
1.1 Motivación .....	17
1.2 Objetivos .....	17
2. ESTADO DEL ARTE.....	18
2.1 Visión histórica.....	18
2.2 Concepto de procesamiento del lenguaje natural .....	19
2.2.1 Fundamentos del Análisis Morfológico .....	20
2.2.2 Etiquetas morfológicas .....	21
2.2.3 Corpus morfológico.....	22
2.2.4 Ambigüedad.....	23
2.3 Etiquetador morfológico.....	23
2.3.1 Etiquetado morfológico manual .....	24
2.3.2 Etiquetado morfológico automático.....	26
2.3.3 Etiquetado morfológico mixto .....	27
2.4 Técnicas de Análisis Morfológico.....	28
2.5 Etiquetas utilizadas en Brown Corpus .....	30
2.6 Desambiguación automática .....	48
2.7 Analizadores POST .....	49
2.8 Herramienta de minería de datos: Aplicación Weka .....	51
2.8.1 Formato de entrada a la aplicación Weka: ARFF .....	57
2.9 Máquina Virtual de Java: Oracle JRockIt.....	58
3. METODOLOGÍA .....	61
3.1 Primera Fase de Metodología.....	61
3.2 Segunda Fase de Metodología.....	63
3.3 Consideraciones con el Corpus .....	65
4. DESARROLLO .....	68
4.1 Aplicación desarrollada: ArffGenerator.....	68
4.2 Algoritmos de Clasificación de Weka.....	75

5.	EXPERIMENTACIÓN 0: Experimentos previos.....	77
6.	EXPERIMENTACIÓN 1: Brown corpus, vocablos ingleses .....	79
6.1	Primera Fase de Experimentación .....	79
6.1.1	Experimento 1.....	79
6.1.2	Experimento 2.....	81
6.1.3	Experimento 3.....	82
6.1.4	Experimento 4.....	83
6.1.5	Experimento 5.....	84
6.1.6	Experimento 6.....	85
6.1.7	Experimento 7.....	86
6.1.8	Experimento 8.....	87
6.1.9	Experimento 9.....	88
6.1.10	Experimento 10.....	89
6.1.11	Experimento 11.....	90
6.1.12	Experimento 12.....	91
6.1.13	Conclusiones y resultados de la Primera Fase de Experimentación.....	92
6.2	Segunda Fase de Experimentación .....	93
6.2.1	Experimentos con número de regla sólo para la palabra (F2-1R) .....	93
6.2.1.1	Experimento 1.....	93
6.2.1.2	Experimento 2.....	94
6.2.1.3	Experimento 3.....	95
6.2.1.4	Experimento 4.....	96
6.2.1.5	Experimento 5.....	97
6.2.1.6	Experimento 6.....	98
6.2.1.7	Experimento 7.....	99
6.2.1.8	Experimento 8.....	100
6.2.1.9	Experimento 9.....	101
6.2.1.10	Experimento 10.....	102
6.2.1.11	Experimento 11.....	103
6.2.1.12	Experimento 12.....	104
6.2.1.13	Conclusiones y resultados sobre la segunda fase (F2-1R) .....	105
6.2.2	Experimentos con número de regla para la palabra y contexto (F2-3R) .....	106
6.2.2.1	Experimento 1.....	106
6.2.2.2	Experimento 2.....	107

6.2.2.3	Experimento 3.....	108
6.2.2.4	Experimento 4.....	109
6.2.2.5	Experimento 5.....	110
6.2.2.6	Experimento 6.....	111
6.2.2.7	Experimento 7.....	112
6.2.2.8	Experimento 8.....	113
6.2.2.9	Experimento 9.....	114
6.2.2.10	Experimento 10.....	115
6.2.2.11	Experimento 11.....	116
6.2.2.12	Experimento 12.....	117
6.2.2.13	Conclusiones sobre la segunda fase (F2-3R).....	118
7.	EXPERIMENTACIÓN 2: Brown corpus, vocablos foráneos.....	119
7.1	Primera Fase de Experimentación.....	119
7.1.1	Experimento 1.....	119
7.1.2	Experimento 2.....	121
7.1.3	Experimento 3.....	122
7.1.4	Experimento 4.....	123
7.1.5	Experimento 5.....	124
7.1.6	Experimento 6.....	125
7.1.7	Experimento 7.....	126
7.1.8	Experimento 8.....	127
7.1.9	Experimento 9.....	128
7.1.10	Experimento 10.....	129
7.1.11	Experimento 11.....	130
7.1.12	Experimento 12.....	131
7.1.13	Conclusiones de la Primera Fase de Experimentación.....	132
7.2	Segunda Fase de Experimentación.....	133
7.2.1	Experimentos con número de regla sólo para la palabra (F2-1R).....	133
7.2.1.1	Experimento 1.....	133
7.2.1.2	Experimento 2.....	134
7.2.1.3	Experimento 3.....	135
7.2.1.4	Experimento 4.....	136
7.2.1.5	Experimento 5.....	137
7.2.1.6	Experimento 6.....	138

7.2.1.7	Experimento 7.....	139
7.2.1.8	Experimento 8.....	140
7.2.1.9	Experimento 9.....	141
7.2.1.10	Experimento 10.....	142
7.2.1.11	Experimento 11.....	143
7.2.1.12	Experimento 12.....	144
7.2.1.13	Conclusiones sobre la segunda fase (F2-1R).....	145
7.2.2	Experimentos con número de regla para la palabra y contexto (F2-3R) .....	146
7.2.2.1	Experimento 1.....	146
7.2.2.2	Experimento 2.....	147
7.2.2.3	Experimento 3.....	148
7.2.2.4	Experimento 4.....	149
7.2.2.5	Experimento 5.....	150
7.2.2.6	Experimento 6.....	151
7.2.2.7	Experimento 7.....	152
7.2.2.8	Experimento 8.....	153
7.2.2.9	Experimento 9.....	154
7.2.2.10	Experimento 10.....	155
7.2.2.11	Experimento 11.....	156
7.2.2.12	Experimento 12.....	157
7.2.2.13	Conclusiones sobre la segunda fase (F2-3R).....	158
8.	RESULTADOS .....	159
9.	PLANIFICACIÓN .....	163
9.1	Estimación de tiempo .....	163
9.2	Estimación de costes.....	166
10.	ENTORNO DE DESARROLLO .....	168
11.	GUIA DEL USUARIO .....	170
12.	CONCLUSIONES.....	172
13.	TRABAJOS FUTUROS .....	173
14.	ANEXOS.....	174
14.1	Guía rápida para utilización de la herramienta Weka .....	174
	GLOSARIO .....	177
	REFERENCIAS .....	181







# ÍNDICE DE FIGURAS

Figura 1: Ejemplo Lematización con lematizador de CLIC .....	20
Figura 2: Ejemplos de análisis morfológico utilizando ENGTWOL .....	23
Figura 3: Ejemplo de transductor .....	29
Figura 4: Pantalla de inicio de Weka .....	54
Figura 5: Ventana del modo Explorador de Weka. Parte de Preprocesado .....	54
Figura 6: Pantalla del modo Explorador de Weka. Parte de Clasificación .....	57
Figura 7: Ejemplo de conversión del diccionario .....	62
Figura 8: Fragmento del fichero de entrada a Weka para la 1ª fase de experimentación.....	62
Figura 9: Proceso de etiquetado con contexto.....	63
Figura 10: Fragmento del fichero de entrada a Weka para la 2ª fase de experimentación con sólo número de regla para la palabra a desambiguar.....	64
Figura 11: Fragmento del fichero de entrada a Weka para la 2ª fase de experimentación con número de regla para la palabra y su contexto.....	64
Figura 12: Frágmento del fichero de correspondencia de palabras compuestas .....	67
Figura 13: Esquema de arquitectura del sistema .....	68
Figura 14: Diagrama de componentes de la aplicación.....	69
Figura 15: Diagrama de casos de uso .....	70
Figura 16: Diagrama de actividades del Menú Principal .....	70
Figura 17: Diagrama de actividad de la creación del Diccionario .....	71
Figura 18: Diagrama de clases para la generación del diccionario .....	71
Figura 19: Diagrama de actividad de la creación del fichero de la Fase 1 de Experimentación.....	72
Figura 20: Diagrama de clases para la generación del fichero de la Fase 1 de experimentación ....	73
Figura 21: Diagrama de actividad de la creación del fichero de la Fase 2 de Experimentación.....	74
Figura 22: Diagrama de clases para la generación del fichero de la Fase 2 de experimentación (5 atributos) .....	74
Figura 23: Diagrama de clases para la generación del fichero de la Fase 2 de experimentación (7 atributos) .....	75
Figura 24: Gráfica comparativa de los algoritmos con más éxito en los experimentos previos .....	78
Figura 25: Gráfica de comparación de los algoritmos en la 1ª fase de experimentación .....	92
Figura 26: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con una regla .....	105
Figura 27: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con tres reglas.....	118

Figura 28: Gráfica de comparación de los algoritmos en la 1ª fase de experimentación .....	132
Figura 29: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con una regla .....	145
Figura 30: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con tres reglas.....	158
Figura 31: Gráfica de comparación entre experimentos realizados.....	159
Figura 32: Comparativa de los algoritmos con mayor acierto en los experimentos previos con los mismos realizados sobre la totalidad de Brown Corpus.....	162
Figura 33: Diagrama de Gantt, Parte 1 .....	164
Figura 34: Diagrama de Gantt, Parte 2 .....	165
Figura 35: Menú principal de la aplicación .....	170
Figura 36: Pantalla de la opción 1, creación del diccionario.....	170
Figura 37: Pantalla de la opción 2, generación del fichero para Fase 1 de Experimentación .....	171
Figura 38: Pantalla de la opción 3, generación del fichero para Fase 2 de Experimentación .....	171
Figura 39: Comando de llamada a Weka .....	174
Figura 40: Pantalla de preprocesamiento de Weka .....	175
Figura 41: Pantalla de clasificación de Weka.....	175



## ÍNDICE DE TABLAS

Tabla 1: Ejemplos de reglas de reconocimiento morfológico en inglés .....	29
Tabla 2: Etiquetas de Adjetivos .....	30
Tabla 3: Etiquetas de Adverbios .....	32
Tabla 4: Etiquetas de Auxiliares .....	34
Tabla 5: Etiquetas de Determinantes .....	36
Tabla 6: Etiquetas de Conjunciones .....	36
Tabla 7: Etiquetas de Sustantivos .....	38
Tabla 8: Etiquetas de Preposiciones .....	38
Tabla 9: Etiquetas de Pronombres .....	41
Tabla 10: Etiquetas de Cualificadores .....	42
Tabla 11: Etiquetas de Cuantificadores .....	42
Tabla 12: Etiquetas de Verbos .....	43
Tabla 13: Etiquetas de Préstamos .....	47
Tabla 14: Etiquetas de Símbolos .....	47
Tabla 15: Resultados de los experimentos .....	162
Tabla 16: Tabla de costes del proyecto .....	167

## RESUMEN

Poder crear sistemas capaces de analizar, entender y responder correctamente en el lenguaje que utilizamos los humanos es una de las grandes metas de muchos investigadores en el campo de la lingüística computacional. Para ello, es necesario que las máquinas sepan procesar el lenguaje natural. Uno de los principales escollos con los que se ha encontrado el procesamiento del lenguaje natural es la ambigüedad de palabras que, aun escribiéndose de la misma manera, pueden tener diferentes significados o desempeñar distintas funciones dentro de una frase dependiendo del contexto en el que se encuentre.

En este proyecto se ha trabajado en la desambiguación morfológica de palabras en inglés aportando la mínima información procedente del contexto. Los textos utilizados para llevar a cabo la investigación proceden de publicaciones en prensa de todo tipo, con lo que se consigue una amplia muestra de la lengua anglosajona. Los términos tienen asociada su categoría gramatical, de manera que lo que se busca son reglas que permitan a procesos de aprendizaje, asignar correctamente a cada palabra su categoría.

Los experimentos que se realizan en este proyecto se pueden clasificar en dos fases. En la primera, se trata de encontrar la categoría de las palabras sin obtener ninguna información del contexto. En la segunda, se utilizan los resultados de la primera fase y las palabras contiguas del vocablo del que se pretende hallar su categoría gramatical.

Los resultados en la segunda fase de experimentación han sido mejores que los de la primera, ya que cuentan con mayor información.



# ESTRUCTURA DEL DOCUMENTO

## Introducción

En esta sección inicial se trata de contextualizar el proyecto “Técnicas de etiquetado y desambiguación morfológica del inglés con reducida información contextual”, describiendo su motivación y objetivos a conseguir, además de explicar brevemente algunos antecedentes que son importantes para el posterior entendimiento del tema central a modo de estado del arte.

## Metodología y Resultados

Se ha seleccionado un corpus en lengua inglesa etiquetado morfológicamente siguiendo las directrices llevadas a cabo en investigaciones y aplicaciones del grupo de investigación *Knowledge Reuse* (KR) de la Universidad Carlos III de Madrid y, posteriormente, se aplican diferentes algoritmos. El corpus en cuestión es el llamado Brown Corpus<sup>1</sup>. Una vez efectuadas las correspondencias entre las etiquetas gramaticales, se describen las acciones realizadas para la generación de las reglas de desambiguación, los experimentos ejecutados y los resultados obtenidos en los mismos.

## Planificación y Entorno de trabajo

Se describe a grandes rasgos las estimaciones, en tiempo y coste, del desarrollo del proyecto. De igual manera, se detalla cual ha sido el entorno de trabajo para la realización de este proyecto.

## Guía del usuario

En este apartado se ha generado una breve guía de usuario para la ejecución de la aplicación desarrollada para la consecución de algunos puntos del proyecto.

## Conclusiones y Trabajos futuros

En este punto se explican las conclusiones obtenidas fruto de la experimentación y se especifican futuros trabajos que se pueden realizar en relación con este proyecto, la precisión del etiquetado y desambiguación morfológica.

## Anexos

En este último punto se explica brevemente la herramienta Weka, que forma parte del proyecto y ha sido de mucha utilidad para la consecución del mismo.

---

<sup>1</sup> Brown Corpus <http://www.hit.uib.no/icame/brown/bcm.html>





# 1. INTRODUCCIÓN

## 1.1 Motivación

La Inteligencia Artificial (IA) es una de las disciplinas modernas en las que más estudios científicos se dan lugar, pues siempre el hombre ha tenido gran inquietud por comprender los procesos de razonamiento y adaptación a cambios. Una de sus áreas más importantes es la lingüística computacional, campo transversal también de la lingüística y la informática, que tiene como objetivo el estudio del lenguaje humano y modelarlo desde un punto de vista computacional. Dentro de las investigaciones que se llevan a cabo en este ámbito intervienen los corpus lingüísticos, etiquetadores, traducciones automáticas o analizadores sintácticos.

El procesamiento del lenguaje natural (abreviado PLN, aunque suele encontrarse como NLP, en inglés *Natural Language Processing*) es otro de los terrenos que aborda la lingüística computacional, y se caracteriza por tener un enfoque esencialmente práctico, orientado al desarrollo de aplicaciones informáticas capaces de entender el lenguaje humano, ya sea oral u escrito. Un claro ejemplo de ello son los traductores automáticos o motores de búsqueda en internet.

Se continúa así con la investigación del grupo *Knowledge Reuse (KR)* de la Universidad Carlos III de Madrid, pues ya tenían previamente una herramienta PLN en inglés. En este caso, en este Proyecto de Fin de Carrera (PFC) se aplican nuevas variantes a la metodología con el fin de categorizar gramaticalmente documentos de escaso texto, como pueden ser facturas o plantillas, donde siempre es importante, sobre todo, identificar los sustantivos y verbos por tener mayor carga semántica. Por lo tanto, el presente PFC se centrará en los procesos de etiquetación y desambiguación morfológica en el idioma inglés.

## 1.2 Objetivos

El principal propósito de este PFC es la desambiguación morfológica de textos en inglés procedentes de textos publicados. Las muestras de texto están recopiladas en un corpus y cada una de las palabras que lo componen está etiquetada con su categoría gramatical.

La desambiguación se va a intentar llevar a cabo con la mínima información contextual, de manera que el etiquetado y el mapeo de dichas etiquetas será lo que marque el devenir en este proyecto.

Se pretende también que el resultado que se obtenga en los experimentos ayude al grupo a progresar en sus estudios en este campo, aportando estadísticas e información relevante contra la que poder confrontar los datos que ya poseen fruto de su propia investigación y de otros proyectos anteriores.

## 2. ESTADO DEL ARTE

El procesamiento del lenguaje natural (PLN), es un área de investigación en continuo desarrollo. Se aplica en la actualidad en diferentes actividades como son los sistemas de recuperación de información, la traducción automática, elaboración automática de resúmenes, mecanismos para comunicación entre personas y máquinas por medio de lenguajes naturales, etc. Aunque en los últimos años se han realizado avances espectaculares, los fundamentos teóricos del PLN se encuentran todavía en estado de desarrollo.

Aún teniendo en cuenta que los obstáculos a superar en el estudio del tratamiento del lenguaje son bastante considerables, los resultados que se van obteniendo y la evolución que se está produciendo en los últimos años sitúan al PLN en posición para liderar una nueva dimensión en las aplicaciones informáticas del futuro: los medios de comunicación del usuario con el ordenador pueden ser más flexibles y el acceso a la información almacenada más eficiente. Por poner un ejemplo, con la creación de interfaces inteligentes, el usuario podría disponer de la facilidad para interactuar con el ordenador en lenguaje natural.

De la misma manera, el uso de técnicas de PLN puede tener un gran impacto en la gestión documental y en los sistemas de traducción automática. En PLN se realizan varias fases de reconocimiento y análisis del lenguaje como la tokenización, la categorización o POS (*Part Of Speech*), la desambiguación, análisis sintácticos, etc. No obstante, la complejidad implícita en el tratamiento del lenguaje comporta limitaciones en los resultados y, por tanto, aplicaciones en áreas de conocimiento concretas y con un uso restringido del lenguaje. Así mismo, el procesamiento del lenguaje natural propicia mejoras en los sistemas de recuperación de información, para la extracción de información y las respuestas a preguntas.

### 2.1 Visión histórica

Las primeras aplicaciones del PLN se dieron durante el período de 1940-1960, teniendo como principal interés la traducción automática. Los experimentos en este sector, basados en la sustitución de palabra por palabra, obtuvieron resultados mediocres y poco esperanzadores.

Surgió, por tanto, la necesidad de resolver ambigüedades sintácticas y semánticas, y por ende, la consideración de información contextual. La ausencia de un orden de la estructura de las oraciones en algunos idiomas, y los problemas para obtener representaciones sintáctica y/o semánticas, constituían los problemas más relevantes. Una vez se afrontaron dichos problemas, se dio paso a una concepción más realista del lenguaje, pues ya se contemplaban las transformaciones que se producen en la estructura de la frase durante el proceso de traducción.

En los años sesenta los intereses se desplazan hacia la comprensión del lenguaje. La mayor parte del trabajo realizado en este período se centró en técnicas de análisis sintáctico.

Hacia los setenta la influencia de los trabajos en inteligencia artificial fue decisiva, centrando su interés en la representación del significado. Como resultado se construyó el primer sistema de preguntas-respuestas basado en lenguaje natural. De esta época data *Eliza*, que era capaz de emular la habilidad de un psicólogo dentro de una conversación. Para ello recogía patrones de información de las respuestas del cliente y elaboraba preguntas que simulaban una discusión.

Entre los años 70 y 80, una vez superados los primeros experimentos, se hacen numerosos intentos de construir programas más fiables. Aparecen gran cantidad de gramáticas con una clara orientación al tratamiento computacional, y se experimenta un notable crecimiento de la tendencia hacia la programación lógica.

En Europa surgen intereses en la elaboración de programas para la traducción automática. Se crea el proyecto de investigación *Eurotra*, que tenía como finalidad la traducción multilingüe. En Japón aparecen equipos dedicados a la creación de productos de traducción para su distribución comercial.

Los últimos años se caracterizan por la incorporación de técnicas estadísticas y se desarrollan formalismos adecuados para el tratamiento de la información léxica. Se introducen nuevas técnicas de representación del conocimiento cercanas a la inteligencia artificial, y las técnicas de procesamiento utilizadas por investigadores procedentes del área de la lingüística e informática son cada vez más próximas. Surgen así mismo, intereses en la aplicación de estos avances en sistemas de recuperación de información con el objetivo de mejorar los resultados en consultas a texto completo.

## **2.2 Concepto de procesamiento del lenguaje natural**

El PLN se concibe como el reconocimiento y utilización de la información expresada en lenguaje humano utilizando sistemas informáticos. En su estudio intervienen diferentes disciplinas tales como lingüística, ingeniería informática, filosofía, matemáticas y psicología. Debido a las diferentes áreas del conocimiento que participan, la aproximación al lenguaje en esta perspectiva es también estudiada desde la llamada ciencia cognitiva. Tanto desde un enfoque computacional como lingüístico, se utilizan técnicas de inteligencia artificial: modelos de representación del conocimiento y de razonamiento, lenguajes de programación declarativos, algoritmos de búsqueda y estructuras de datos.

Se investiga cómo se puede aprovechar el lenguaje para satisfacer gran número de tareas y como modelar el conocimiento. Para entender y atacar lo planteado, hay que tener en cuenta la doble dimensión que suscita el procesamiento del lenguaje natural: se trata por una parte de un problema de representación lingüística, y por otra de un problema de tratamiento mediante recursos informáticos.

El uso de técnicas computacionales (procedentes especialmente de la inteligencia artificial) no arrojaría soluciones adecuadas sin una concepción profunda del fenómeno lingüístico. Por otra parte, las gramáticas utilizadas para el tratamiento del lenguaje han evolucionado hacia modelos más adecuados para un tratamiento computacional. El estudio del lenguaje natural se estructura normalmente en 4 niveles de análisis:

- Morfológico
- Sintáctico
- Semántico
- Pragmático.

Además, se pueden incluir otros niveles de conocimiento como es la información fonológica, referente a la relación de las palabras con el sonido asociado a su pronunciación; el análisis del discurso, que estudia cómo la información precedente puede ser relevante para la comprensión de otra información; y, finalmente, lo que se denomina conocimiento del mundo, referente al conocimiento general que los hablantes han de tener sobre la estructura del mundo para mantener una conversación. En este PFC nos centramos en el Análisis Morfológico, por lo que de ahora en adelante nos referiremos a ese nivel en exclusiva.

### 2.2.1 Fundamentos del Análisis Morfológico

La función básica del Analisis Morfológico consiste en detectar la relación que se establece entre las unidades mínimas que forman una palabra, como puede ser el reconocimiento de sufijos o prefijos, y darles una categoría gramatical. Este nivel de análisis mantiene una estrecha relación con el léxico.

El léxico es el conjunto de información sobre cada palabra que el sistema utiliza para el procesamiento. Las palabras que forman parte del diccionario están representadas por una entrada léxica, y en caso de que ésta tenga más de un significado o diferentes categorías gramaticales, tendrá asignada diferentes entradas. En el léxico se incluye la información morfológica, la categoría gramatical, irregularidades sintácticas y representación del significado.

Normalmente el léxico sólo contiene la raíz (o lema) de las palabras con formas regulares, siendo el analizador morfológico el que se encarga de determinar si el género, número o flexión que componen el resto de la palabra son adecuados.



Figura 1: Ejemplo Lematización con lematizador de CLIC

El Análisis Morfológico en PLN consta de dos elementos fundamentales: las etiquetas que se establecen para el análisis y los documentos que se van a utilizar. En este caso, dichos documentos conforman un conjunto de información digitalizada y recursos denominado corpus morfológico. Ese corpus morfológico constituye un lexicón etiquetado, es decir, un claro ejemplo del vocabulario del idioma en cuestión, listando los diferentes elementos léxicos que componen esa determinada lengua con sus correspondientes etiquetas.

### 2.2.2 Etiquetas morfológicas

Las categorías gramaticales se pueden clasificar en categorías cerradas (o menores) y categorías abiertas (o mayores) atendiendo al número de palabras que las componen. Las categorías cerradas son aquellas que tienen un número fijo de palabras (verbos auxiliares, conjunciones, determinantes, artículos, preposiciones). Las categorías abiertas son aquellas potencialmente infinitas y variables en número (verbos, sustantivos, adjetivos, adverbios).

En inglés, se han establecido las siguientes categorías gramaticales para este PFC:

1. sustantivos (nouns): cat, car, book, etc.
2. verbos (verbs): run, win, open, etc.
3. adjetivos (adjectives): beautiful, clean, blue, etc.
4. adverbios (adverbs): yes, no, now, really, etc.
5. cualificadores (qualifiers): quite, rather, enough, etc.
6. pronombres (pronouns): I, you, they, me, etc.
7. preposiciones (prepositions): in, at, on, etc.
8. determinantes (determiners): that, this, many, etc.
9. cuantificadores (quantifiers): half, both, all, etc.
10. conjunciones (invariant): but, and, if, etc.
11. préstamos (foreignword): cualquier palabra de otra lengua.
12. auxiliares (aux): be, can, will, etc.

No obstante, los juegos de etiquetas gramaticales, dependiendo de su aplicación, pueden tener otras categorías más específicas jerarquizadas. La desambiguación sobre el abanico de categorías candidatas específicas (en este caso, no se tiene conocimiento de las categorías candidatas de las palabras a desambiguar) de un término es también posible a través de la desambiguación de sus categorías gramaticales genéricas. Por ejemplo, si las categorías candidatas del término “fly” son “nombre común” y “verbo en presente de indicativo para cualquier persona excepto la 3ª de singular”, para desambiguar a una de ellas sería suficiente saber si la palabra es un sustantivo o un verbo.

Algunas palabras pueden variar su forma dependiendo de las construcciones sintácticas de que forman parte<sup>2</sup>. Atendiendo a esta característica, las categorías

---

<sup>2</sup> En lingüística, la flexión o inflexión es la modificación de una palabra para expresar diferentes categorías gramaticales como el tiempo, modo, voz, aspecto, persona, número, género y caso. La flexión de los verbos también se llama conjugación, y la inflexión de los sustantivos, adjetivos y pronombres también se llama declinación.

gramaticales también se puede clasificar en dos grupos: las variables y las invariables. Las categorías gramaticales o partes de la oración variables son aquellas que tienen accidentes gramaticales (o morfemas flexivos). A continuación se muestran los accidentes gramaticales del inglés.

**NOMINALES:** afectan al artículo, al sustantivo, al adjetivo y al pronombre.

- Género: Masculino y femenino
- Número: Singular y plural
- Caso: Sujeto, objeto, posesivo y reflexivo

**VERBALES:** afectan al verbo.

Voces: Activa y Pasiva

Modos:

- Personales: Indicativo, Subjuntivo e Imperativo
- Impersonales (formas nominales): participio, infinitivo y gerundio

Tiempos:

- Simples: Presente, Pasado y Futuro.
- Continuo: Presente, Pasado y Futuro.
- Perfecto: Presente, Pasado y Futuro.
- Perfecto Continuo: Presente, Pasado y Futuro.

Número y personas:

- Singular: 1ª (I), 2ª (you), 3ª (he, she, it)
- Plural: 1ª (we), 2ª (you), 3ª (they)

### 2.2.3 Corpus morfológico

La utilización de los corpus aplicados al PLN se ha venido acrecentando en los últimos años, de manera que se han establecido como un punto de apoyo casi esencial. Son empleados en gran cantidad de tareas, tales como creación de supuestos, análisis lingüísticos o sólidos sistemas de PLN.

El corpus es una colección de textos codificados electrónicamente, que pueden proceder del lenguaje oral, escrito o de ambos, clasificados y ordenados, disponibles para servir de base a investigaciones.

Para la creación de un corpus morfológico se codifican las características morfológicas que comporten un significado gramatical específico. Por ejemplo, la persona y el número para la categoría verbal. Esta información comporta relaciones de concordancia así como la que se establece entre los sustantivos y adjetivos. Por otra parte también se codifican rasgos léxicos y de posición de palabras como determinante o adjetivo antepuesto. Este tipo de codificación añadida a la morfológica es la que incluyen los corpus ingleses como el Brown Corpus (el utilizado en este PFC) o el Penn TreeBank<sup>3</sup>.

El Brown Corpus fue preetiquetado automáticamente por el etiquetador TAGGIT [Barbara B. Greene y Gerald M. Rubin, Universidad de Brown, Providence 1971]. Esta herramienta utilizaba la información léxica para limitar las etiquetas de las palabras y aplicaba las reglas de etiquetado cuando las palabras del contexto no tenían

---

<sup>3</sup> Penn Treebank <http://www.cis.upenn.edu/~treebank/>

ambigüedades. Una vez se obtuvo la salida del etiquetador, fue corregido manualmente con gran esfuerzo personal. Para este PFC se ha dispuesto de una versión del Brown Corpus etiquetada por el *tagger* AMALGAM.

### 2.2.4 Ambigüedad

El principal problema al que se enfrentan los procesadores del lenguaje natural es la ambigüedad. La ambigüedad puede venir dada como polisemia (existencia de morfemas que pueden tener varios significados) u homografía (existencia de varias palabras totalmente diferentes que se escriben igual). En el caso que nos atañe, dentro del análisis morfológico la ambigüedad consistiría en que una de las palabras pudiera pertenecer a dos o más categorías gramaticales. Por ejemplo, la palabra *wind* puede ser un sustantivo o un verbo.

## 2.3 Etiquetador morfológico

Un etiquetador morfológico basa su funcionamiento en buscar la forma canónica de la palabra a clasificar, su categoría gramatical y la flexión que hay en ella.

The figure displays three sequential screenshots of a web-based morphological analysis tool. Each screenshot shows a text input field for a word, an 'Analyse' button, and the resulting morphological analysis output.

**First Screenshot (Word: cars):**

```
The analysis of the word cars (see the description of morphological tags and other notations):  
"<cars>"  
    "car" N NOM PL
```

**Second Screenshot (Word: stopped):**

```
The analysis of the word stopped (see the description of morphological tags and other notations):  
"<stopped>"  
    "stop" <SV0> <SV> V PAST VFIN @+FMAINV  
    "stop" <SV0> <SV> PCP2
```

**Third Screenshot (Word: probably):**

```
The analysis of the word probably (see the description of morphological tags and other notations):  
"<probably>"  
    "probable" <DER:bly> ADV
```

Figura 2: Ejemplos de análisis morfológico utilizando ENGTWOL

Debido a que el inglés no es un idioma en el que la flexibilidad sea muy relevante, en la manera de etiquetar las palabras (y sobretodo su éxito y fiabilidad) tiene mayor relevancia ubicarla en el contexto o entorno de la misma dentro de la oración que

tratarlas de manera aislada. Todo lo contrario que ocurre en el castellano, ya que su mayor riqueza flexiva permite identificar con mayor facilidad su clase gramatical.

Fundamentalmente existen dos maneras diferentes de etiquetación de palabras, diferenciadas por la utilización o no de herramientas informáticas. Es por ello que se pueden agrupar en etiquetadores morfológicos automáticos, etiquetadores morfológicos manuales y etiquetadores morfológicos mixtos.

### 2.3.1 Etiquetado morfológico manual

Mediante la aplicación de etiquetas o *tags* se puede enriquecer el texto que componen los corpus con informaciones estructurales, semánticas o de otra índole. Este laborioso proceso normalmente es llevado a cabo por un grupo de analistas para poder afinar todo lo posible en las anotaciones y así incurrir en el menor número de errores tras sucesivas pruebas y revisiones.

Para realizar un etiquetado correcto se deben tener en cuenta ciertas normas de anotación, que deben ser conocidas y accesibles, y que además pueden contener errores. También debe seguir alguno de los estándares de codificación conocidos:

- **TEI** (*Text Incode Initiative*): Este estándar se divide en dos partes: una descripción textual discursiva con ejemplos extensos y un conjunto de definiciones de etiquetas. Los esquemas en la mayoría de los formatos modernos (DTD , Relax NG y esquema del W3C ) se generan automáticamente a partir de las definiciones de etiquetas. Está disponible en XML.
- **CES** (*Corpus Encoding Standard*): Se ha desarrollado para especificar un nivel de codificación mínima para que un corpus pueda ser considerado normalizado en términos de la representación descriptiva (marcado de información estructural y tipográfica), así como la arquitectura general (con el fin de ser adecuado para su uso en una base de datos de texto). También proporciona codificación específica para la anotación lingüística, junto con una arquitectura de datos de corpus lingüísticos. Se encuentra disponible en XML (llamado *XCES*) y SGML.
- **LDC** (*Linguistic Data Consortium*): Está compuesto por bases de datos de texto con más de un millón de palabras y en la mayoría de los idiomas. Los textos son normalizados por la conversión de su codificación de caracteres en un formulario estándar y la inserción de un formulario estándar de SGML marcado. La mayor parte de las adquisiciones de texto de LDC han sido financiadas a cargo de proyectos de modelado de lenguaje para el reconocimiento de voz, recuperación de información y la comprensión de mensajes y de bases de datos diseñadas para apoyar la enseñanza de idiomas. También fomenta la creación de corpus para idiomas que actualmente carecen de tales recursos.
- **ELRA** (*European Languages Resources Association*): La misión de esta asociación europea es promover los recursos lingüísticos y de evaluación para el sector de las Tecnologías del Lenguaje Humano en todas sus formas y usos. En



consecuencia, los objetivos son: coordinar y llevar a cabo la identificación, producción, validación, distribución y estandarización de los recursos idiomáticos, así como el apoyo para la evaluación de los sistemas y herramientas para su explotación. ELDA (Evaluations and Language resources Distribution Agency) forma parte de ELRA, y se encarga de las cuestiones prácticas y legales relacionadas con la distribución de los recursos lingüísticos, siendo la encargada de tomar decisiones y acuerdos de distribución en nombre de ELRA.

- **EAGLES** (Expert Advisory Group on Language Engineering Standards): Iniciativa de la Comisión Europea, dentro del programa de Investigación e Ingeniería del Lenguaje del antiguo DG XIII, que tiene como objetivo acelerar la provisión de normas para recursos lingüísticos a gran escala (por ejemplo, corpus) y creación de herramientas de manipulación y evaluación para ese conocimiento. Empresas, universidades y centros de investigación colaboran desarrollando las directrices de EAGLES y así tratan de conformar unas normas de buena práctica en las principales áreas de la ingeniería lingüística.

El proceso de etiquetado es bastante más complicado de lo que a priori podría parecer. Se han seguido varias técnicas para aumentar su precisión y fiabilidad. Una de ellas es utilizar gramáticas de restricciones. Los primeros sistemas de etiquetado basados en reglas constaban de dos fases. En la primera de ellas se conformaba un diccionario que asignaba a cada palabra un listado de todas las etiquetas posibles para dicha palabra y en la segunda se disponía de una lista de reglas de desambiguación escritas a mano para así lograr que a cada palabra se le asigne una sola etiqueta. Este concepto (*Constraint Grammar* o *CG*) fue mostrado y desarrollado principalmente por Fred Karlsson entre 1990 y 1995, y hacía posible la desambiguación morfológica teniendo en cuenta todas sus posibles interpretaciones.

Los fundamentos de las gramáticas de restricciones establecen una serie de reglas dependientes del contexto que asignan etiquetas gramaticales a las palabras y símbolos del texto. Cada una de esas reglas añade, elimina, selecciona o reemplaza una etiqueta o conjunto de etiquetas de un determinado contexto de la frase. Estas condiciones del contexto se pueden vincular a cualquier etiqueta o conjunto de etiquetas de cualquier palabra y frase. Incluso las reglas pueden estar vinculadas o condicionadas unas a otras, negándolas o bloqueándolas.

Normalmente, se crean miles de reglas que se van aplicando en pasos progresivos, cubriendo niveles cada vez más avanzados de análisis. En cada nivel, las normas de seguridad se utilizan antes que las heurísticas, y no son permitidas las reglas para eliminar las últimas etiquetas que queden de ese tipo, pues de esa manera se consigue mayor grado de robustez en la solución. Al ser un proceso realizado manualmente por uno o varios humanos, se suele reducir el número de errores.

Los sistemas de gramática de restricciones se pueden utilizar para crear árboles sintácticos complejos en otros formalismos añadiendo pequeñas modificaciones o gramáticas de dependencias. La metodología CG también se ha utilizado en una serie de aplicaciones de tecnología lingüística, como los correctores ortográficos y sistemas de traducción automática.

La precisión en el etiquetado depende en gran medida de las sucesivas decisiones que se vayan tomando a la hora de seleccionar las reglas para evitar la desambiguación, ya que se puede utilizar métodos de actuación de eliminación o de puntuación, lo que plantea una solución variable y subjetiva, también fundamentada en los fines para los que se haga.

### 2.3.2 Etiquetado morfológico automático

Un *tagger* o etiquetador morfológico es una aplicación informática que es capaz de leer textos de una lengua y asignar a cada una de las palabras, la categoría gramatical que le corresponde, teniendo en cuenta el contexto en el que aparece en la frase. Surgen de la práctica imposibilidad de etiquetar grandes textos de manera manual, ya que supondría destinar una gran cantidad de recursos, tanto de tiempo como personales, para conseguir que el proceso fuera fiable.

La investigación sobre el etiquetado POS (acrónimo inglés que proviene de “*Part of Speech*”, cuya traducción es “Parte Del Discurso”) ha estado estrechamente ligada a la lingüística de corpus. El primer corpus importante en inglés para el análisis computacional fue el Brown Corpus desarrollado en la Universidad de Brown por Henry Kucera y Nelson Francis, a mediados de la década de 1960. Se compone de alrededor de un millón de palabras en lengua inglesa, compuesto de 500 muestras de publicaciones elegidas al azar. Cada fragmento es de 2.000 o más palabras, terminando cada uno de ellos al aparecer un punto de final de oración, de manera que el corpus contiene sólo frases completas.

Durante mucho tiempo, el etiquetado POS se ha considerado una parte inseparable de procesamiento del lenguaje natural, porque hay ciertos casos en los que la categoría gramatical de la palabra no puede ser resuelta correctamente sin entender la semántica o incluso la pragmática del contexto. Esto es extremadamente costoso, sobre todo porque el análisis a niveles más altos es mucho más difícil cuando se barajan varias categorías que deben ser consideradas para cada palabra.

El corpus Brown Corpus ha sido objeto de numerosos y laboriosos etiquetados gramaticales durante muchos años. Como primera aproximación, Greene y Rubin desarrollaron una aplicación llamada *Taggit*, que consistía en una enorme lista hecha a mano de las categorías gramaticales en la que la aparición de una podía estar relacionada con la aparición de otra. Por ejemplo, en el orden de una oración, a un artículo le sucede por norma general un sustantivo, pero nunca un adverbio. El programa consiguió un 70 % de etiquetados correctos. Como curiosidad, hay que reseñar que contiene un ejemplo con 17 palabras ambiguas en una fila, y hay palabras tales como “*still*” que puede representar hasta 7 categorías gramaticales diferentes. Sus resultados fueron revisados en varias ocasiones y corregidos a mano, además de que los usuarios finales también reportaban los errores cometidos, por lo que a finales de los años 70 el etiquetado era casi completo y perfecto, acertando algunos casos en los que incluso los humanos podrían no estar totalmente seguros.

Este corpus se ha utilizado para innumerables estudios sobre la frecuencia de palabras y la desambiguación gramatical, y ha inspirado el desarrollo de corpus etiquetados de manera semejante en otros idiomas. Sin embargo, Brown Corpus ha sido reemplazado por corpus más grandes, como el *BNC* (British National Corpus) que consta de 100 millones de palabras.

En la década de los 80, los investigadores en Europa comenzaron a utilizar los HMMs (acrónimo inglés que proviene de “*Hidden Markov Models*”, cuya traducción es “*Modelos Ocultos de Markov*”) para intentar eliminar la ambigüedad surgida en la etiquetación Corpus de Inglés Británico. Los HMMs implican contar los diferentes casos surgidos en el corpus y componer una tabla de las probabilidades de ciertas secuencias. Por ejemplo, una vez detectado un artículo como “*the*”, la siguiente palabra es un sustantivo el 40% de las veces, un adjetivo el 40%, y un número de 20 %. Sabiendo esto, un programa puede decidir que la palabra “*can*” en “*the can*” es mucho más probable que sea un sustantivo que un verbo. El mismo método puede, por supuesto, ser utilizado para beneficiarse de los conocimientos sobre las restantes palabras de la frase y del corpus en general.

Existen Modelos Ocultos de Markov (también llamados modelos de *n-gramas*) más avanzados que aprenden las probabilidades no sólo de parejas de palabras, sino tripletas o secuencias aún más grandes. Así, por ejemplo, tras observar un sustantivo seguido de un verbo, el siguiente elemento puede ser muy probablemente una preposición, un artículo o un sustantivo, pero es muy improbable que sea otro verbo. Cuando varias palabras ambiguas se presentan juntas, las posibilidades se multiplican. Sin embargo, es fácil enumerar todas las combinaciones y asignar una probabilidad relativa a cada una de ellas, multiplicando sucesivamente las probabilidades de cada elección. A continuación, se elige la combinación con mayor probabilidad. El grupo europeo desarrolló *CLAWS*, un programa de marcado que hizo exactamente esto, y que logra una precisión en el rango de 93-95 %. *CLAWS* fue el pionero en el campo del marcado de voz basado en Modelos Ocultos de Markov, pero era realmente costoso, ya que enumeraba todas las posibilidades.

### 2.3.3 Etiquetado morfológico mixto

Los etiquetadores morfológicos mixtos combinan técnicas de etiquetado manual y automáticas en pos de un grado mayor de acierto y menor esfuerzo humano en las tareas. Un problema potencial con etiquetadores basados en *n-gramas* es el tamaño de su tabla de *n-gramas* generada, lo que ocasiona graves problemas computacionales. Es importante lograr un equilibrio entre el tamaño del modelo y el rendimiento etiquetador para que sea viable su uso en computadores. Un etiquetador de *n-gramas* puede almacenar tablas de trigramas y bigramas, grandes matrices dispersas que pueden tener cientos de millones de entradas.

Una segunda cuestión se refiere a su contexto. La única información que un etiquetador de *n-gramas* considera a partir del contexto anterior es las etiquetas, a pesar de que las palabras por sí mismas pueden ser una fuente útil de información. Es muy poco práctico para los modelos de *n-gramas* condicionar las identidades de las palabras del contexto. Un claro ejemplo de etiquetado utilizando este tipo de modelo es el etiquetador de Brill, que utiliza un método de marcado inductivo que se desempeña muy bien utilizando modelos que son sólo una pequeña fracción del tamaño de los etiquetadores de *n-gramas*.

El etiquetador de Brill [Eric Brill, 1995] es un tipo de aprendizaje basado en la transformación continua. La idea general es muy simple: adivinar la etiqueta de cada palabra, para a continuación, volver atrás y corregir los errores. De esta manera, un

etiquetador Brill transforma sucesivamente una etiqueta errónea de un texto en una correcta. Al igual que con los etiquetadores de HMMs, este es un método de aprendizaje supervisado, ya que necesitamos datos de entrenamiento con anotaciones para averiguar si la suposición del etiquetador es correcta o no. Sin embargo, a diferencia de los etiquetadores de n-gramas, no tiene en cuenta las observaciones previas, pero compila una lista de reglas de corrección de transformación que va manteniendo y actualizando a lo largo del proceso.

El proceso del etiquetador Brill suele explicarse por analogía con la pintura. Supongamos que estábamos pintando un árbol, con todos sus detalles de ramas, pequeñas ramas y hojas, contra un fondo de color azul celeste uniforme. En lugar de pintar el árbol primero y luego tratar de pintar de azul los espacios vacíos, es más sencillo pintar todo el lienzo azul, y a continuación, la parte del árbol por el exceso de pintura de fondo azul. De la misma manera, se podría pintar el tronco de un color marrón uniforme antes de volver a pintar más detalles con pinceles aún más finos. Brill utiliza la misma idea: comenzar con pinceladas amplias para luego ir arreglando los pequeños detalles, con cambios cada vez más sutiles.

Aparte del etiquetador Brill, hay otro basado en reglas llamado RDRPOSTagger [Nguyen, D. Q., Nguyen, D. Q., Pham, S. B., y Pham, D. D., 2011] que aplica una metodología de adquisición de conocimientos incrementales donde las reglas se almacenan en una estructura de excepción y las nuevas reglas sólo se añaden para corregir errores en las ya vigentes.

Otros etiquetadores se basan en la utilización de algoritmos dinámicos como el de Viterbi [Andrew James Viterbi, 1965], que se apoya en los Modelos Ocultos de Markov para encontrar la mejor secuencia de estados en una secuencia de eventos observados. Este algoritmo se utiliza, por ejemplo, en aplicaciones de reconocimiento de voz y utilización de texto predictivo en telefonía móvil.

Al igual que ocurre con el algoritmo de Viterbi, existen etiquetadores que intentan explotar algoritmos aplicados a los Modelos Ocultos de Markov. Uno de los más conocidos es el Baum-Welch [Leonard E. Baum y Lloyd R. Welch, 1970], que es capaz de detectar patrones utilizando algoritmos de esperanza-maximización (EM) y así predecir y estimar el valor de los parámetros desconocidos del modelo con mayor semejanza a la realidad. Este algoritmo ha sido utilizado sobre todo para aplicaciones de síntesis de voz.

## **2.4 Técnicas de Análisis Morfológico**

Se pueden distinguir dos tipos de técnicas utilizadas para el análisis morfológico: las basadas en reglas y las de estados finitos.

Las técnicas de análisis morfológico basadas en reglas se caracterizan por tener gran eficiencia y son extensibles, de manera que pueden ampliarse hasta un gran nivel de detalle. Estas técnicas son utilizadas con gran éxito en lenguas con poca variación morfológica, como lo es el inglés. En idiomas muy flexivos, cuando existe gran cantidad

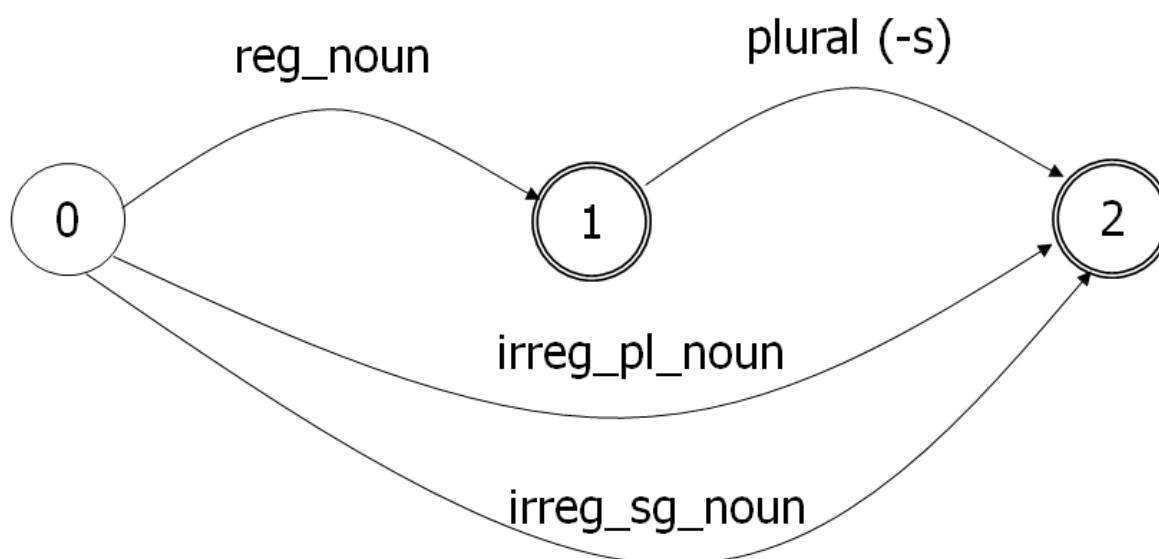
de morfología derivativa (prefijos, sufijos e infijos) en sus palabras, no se consigue tanta fiabilidad. En la siguiente tabla, se muestran algunas reglas que permiten el reconocimiento morfológico del inglés.

<i>Nombre</i>	<i>Descripción</i>	<i>Ejemplo</i>
U después de Q	La letra “q” siempre va seguida de la vocal “u”	Quiz, Squash
E final de palabra	Se elimina la “e” final que no se pronuncia al añadir las terminaciones “ed” o “ing”	Make/Making
LL final de palabra	Las palabras que terminan en doble “ll”, eliminan la “l” final cuando se añaden las terminaciones “ment” o “ful”	Install/Instalment Skill/Skilful
Y final de palabra	Al agregar las terminaciones a las palabras que terminan con una consonante e “y” final, se cambia la “y” por “i” (a menos que los que se añade comience por “i”, como por ejemplo “ish”)	Defy/Defied/Defying

**Tabla 1: Ejemplos de reglas de reconocimiento morfológico en inglés**

En cuanto a las técnicas de análisis basadas en estados finitos, dependiendo de los niveles con los que cuente el analizador pueden utilizar autómatas finitos (solo un nivel) o transductores (dos o más niveles). Jonhson [1982] fue el pionero en observar que ciertas reglas, tanto morfológicas como fonológicas, podían ser representadas por aparatos de estados finitos, dándole el nombre de *modelo de dos niveles*. Más adelante, y apoyándose en las ideas de Johnson, Koskenniemi dio a conocer su modelo de correspondencia entre la forma léxica de las palabras y sus derivaciones en 1983.

Un transductor es un modelo computacional compuesto por un alfabeto, un conjunto de estados y unas determinadas funciones de transición que especifican el cambio de un estado a otro. Las funciones de transición son etiquetadas con un par de símbolos que constituyen el alfabeto del *input* y el alfabeto de *output*. Este mecanismo se puede representar en la forma de un diagrama o gráfico de estado-finito. El transductor tomaría cadenas en el *input* y las relacionaría con cadenas en el *output*.



**Figura 3: Ejemplo de transductor**

## 2.5 Etiquetas utilizadas en Brown Corpus

La versión del Brown Corpus utilizada en este Proyecto Fin de Carrera contiene el etiquetado realizado por Barbara B. Green y Gerald M. Rubin en 1971. Dicho etiquetado tiene poco que ver con otros estándares como por ejemplo el EAGLES, dado que en este caso sí se utilizan etiquetas combinadas, en su versión AMALGAM.

Cualquiera de las palabras negadas tienen un asterisco anexado después de su etiqueta o el símbolo más (“+”) separa las etiquetas para los diferentes tokens que componen la palabra combinada completa. Esto hace que sea una tarea trivial dividir las etiquetas combinadas. De esta forma, lo que se ha hecho durante el experimento es establecer las equivalencias de palabras combinadas en sus partes constituyentes y las etiquetas aplicadas a cada parte. Por lo tanto, “*won’t*” (MD \*) se convierte en “*will*” (MD) y “*not*” (\*); o “*I’d*” (SSPP + HVD) se convierte en “*I*” (SSPP) y “*had*” (HVD).

A continuación se presentan las etiquetas que se han utilizado en este corpus. Para cada categoría se presentan la etiqueta, su descripción y un ejemplo del token al que representa.

### 1. Adjetivos (Adjectives):

Adjetivos		
Etiqueta	Descripción	Ejemplo
<b>JJ</b>	adjetivo	fit
<b>JJ\$</b>	adjetivo, genitivo	Great's
<b>JJ+JJ</b>	adjetivos, par unido por guión	long-far
<b>JJR</b>	adjetivo, comparativo	older
<b>JJR+CS</b>	adjetivo y conjunción coordinada	lighter'n
<b>JJS</b>	adjetivo, semánticamente superlativo	top
<b>JJT</b>	adjetivo, superlativo	best

Tabla 2: Etiquetas de Adjetivos

## 2. Adverbios (Adverbs):

Adverbios		
Etiqueta	Descripción	Ejemplo
*	adverbio, negación	not
<b>RB</b>	adverbio	only
<b>RB\$</b>	adverbio, genitivo	else's
<b>RB+BEZ</b>	adverbio, contracción con "to be", tercera persona de singular del presente	here's
<b>RB+CS</b>	adverbio y conjunción, coordinada	well's
<b>RBR</b>	adverbio, comparativo	longer
<b>RBR+CS</b>	adverbio, comparativo y conjunción coordinada	more'n
<b>RBT</b>	adverbio, superlativo	highest
<b>RN</b>	adverbio, nominal	here
<b>RP</b>	adverbio, particular	up
<b>RP+IN</b>	adverbio particular, contracción con preposición	out'n
<b>WRB</b>	WH-adverbio	why
<b>WRB+BER</b>	WH-adverbio, contracción con "to be", segunda persona de singular o cualquier persona de plural del presente	where're
<b>WRB+BEZ</b>	WH-adverbio, contracción con "to be", tercera persona de singular del presente	how's
<b>WRB+DO</b>	WH-adverbio contracción con "to do", presente excepto tercera persona de singular	howda
<b>WRB+DOD</b>	WH-adverbio, contracción con "to do", pasado	where'd

<b>Adverbios</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>WRB+DOD*</b>	WH-adverbio, contracción con "to do", pasado, negación	whyn't
<b>WRB+DOZ</b>	WH-adverbio contracción con "to do", tercera persona de singular del presente	how's
<b>WRB+IN</b>	WH-adverbio, contracción con preposición	why'n
<b>WRB+MD</b>	WH-adverbio, contracción con auxiliar modal	where'd

**Tabla 3: Etiquetas de Adverbios**

### 3. Auxiliares (Aux)

<b>Auxiliares</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>BE</b>	verbo "to be", infinitivo o imperativo	be
<b>BED</b>	verbo "to be", pasado, segunda persona de singular o cualquier persona de plural	were
<b>BED*</b>	verbo "to be", pasado, segunda persona de singular o cualquier persona de plural, negado	weren't
<b>BEDZ</b>	verbo "to be", pasado, primera y tercera persona de singular	was
<b>BEDZ*</b>	verbo "to be", pasado, primera y tercera persona de singular, negado	wasn't
<b>BEG</b>	verbo "to be", participio presente o gerundio	being
<b>BEM</b>	verbo "to be", presente, primera persona de singular	am



<b>Auxiliares</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>BEM*</b>	verbo "to be", presente, primera persona de singular, negado	ain't
<b>BEN</b>	verbo "to be", participio pasado	been
<b>BER</b>	verbo "to be", presente, segunda persona de singular o cualquier persona de plural	are
<b>BER*</b>	verbo "to be", presente, segunda persona de singular o cualquier persona de plural, negado	aren't
<b>BEZ</b>	verbo "to be", presente, tercera persona de singular	is
<b>BEZ*</b>	verbo "to be", presente, tercera persona de singular, negada	isn't
<b>DO</b>	verbo "to do", presente, infinitivo o imperativo	do
<b>DO*</b>	verbo "to do", presente, infinitivo o imperativo, negado	don't
<b>DO+PPSS</b>	verbo "to do", pasado o presente contracción con pronombre personal nominativo excepto tercera persona de singular	d'you
<b>DOD</b>	verbo "to do", pasado	did
<b>DOD*</b>	verbo "to do", pasado, negado	didn't
<b>DOZ</b>	verbo "to do", presente, tercera persona de singular	does
<b>DOZ*</b>	verbo "to do", presente, tercera persona de singular, negado	doesn't
<b>HV</b>	verbo "to have", presente, infinitivo o imperativo	have
<b>HV*</b>	verbo "to have", presente, infinitivo o imperativo, negado	haven't

<b>Auxiliares</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>HV+TO</b>	verbo "to have", presente contracción con "to"	hafta
<b>HVD</b>	verbo "to have", pasado	had
<b>HVD*</b>	verbo "to have", pasado, negado	hadn't
<b>HVG</b>	verbo "to have", participio presente o gerundio	having
<b>HVN</b>	verbo "to have", participio pasado	had
<b>HVZ</b>	verbo "to have", presente, tercera persona de singular	has
<b>HVZ*</b>	verbo "to have", presente, tercera persona de singular, negado	hasn't
<b>MD</b>	auxiliar modal	will
<b>MD*</b>	auxiliar modal, negado	cannot
<b>MD+HV</b>	auxiliar modal contracción con verbo "to have"	must've
<b>MD+PPSS</b>	auxiliar modal contracción con pronombre personal excepto tercera persona de singular	willya
<b>MD+TO</b>	auxiliar modal contracción con "to"	oughta

**Tabla 4: Etiquetas de Auxiliares**

#### **4. Determinantes (Determiners)**

<b>Determinantes</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>AP</b>	determinante	very
<b>AP\$</b>	determinante, genitivo	other's

<b>Determinantes</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>AP+AP</b>	determinante, par de determinantes unidos por guión	many-much
<b>AT</b>	determinante, artículo	the
<b>CD</b>	determinante numeral, cardinal	two
<b>CD\$</b>	determinante numeral, cardinal, genitive	1960's
<b>DT</b>	determinante pronombre, singular	this
<b>DT\$</b>	determinante pronombre, singular, genitivo	another's
<b>DT+BEZ</b>	determinante pronombre, contracción con verbo "to be", presente, tercera persona de singular	that's
<b>DT+MD</b>	determinante pronombre, contracción con auxiliar modal	this'll
<b>DTI</b>	determinante pronombre, singular o plural	any
<b>DTS</b>	determinante pronombre, plural	them
<b>DTS+BEZ</b>	determinante pronombre plural contracción con verbo "to be", presente, tercera persona de singular	them's
<b>DTX</b>	determinante pronombre	one
<b>OD</b>	determinante numeral, ordinal	third
<b>WDT</b>	WH-determinante	what
<b>WDT+BER</b>	WH-determinante, contracción con verbo "to be", presente, segunda persona de singular o cualquier persona de plural	what're

<b>Determinantes</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>WDT+BER+PP</b>	WH-determinante, contracción con verbo "to be", presente, segunda persona de singular o cualquier persona de plural, contracción con pronombre personal excepto tercera persona de singular	whaddya
<b>WDT+BEZ</b>	WH-determinante, contracción con verbo "to be", presente, tercera persona de singular	what's
<b>WDT+DO+PPS</b>	WH-determinante, contracción con verbo "to do", presente, contracción con pronombre personal excepto tercera persona de singular	whaddya
<b>WDT+DOD</b>	WH-determinante, contracción con verbo "to do", pasado	what'd
<b>WDT+HVZ</b>	WH-determinante, contracción con verbo "to have", presente, tercera persona de singular	what's

**Tabla 5: Etiquetas de Determinantes**

## 5. Conjunciones (Invariants)

<b>Conjunciones</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>CC</b>	conjunción coordinada	and
<b>CS</b>	conjunción subordinada	because
<b>TO</b>	palabra "to"	to
<b>TO+VB</b>	palabra "to" contracción con verbo en infinitivo	t'jawn

**Tabla 6: Etiquetas de Conjunciones**

## 6. Sustantivos (Nouns)

Sustantivos		
Etiqueta	Descripción	Ejemplo
NN	sustantivo común, singular	airport
NN\$	sustantivo común, singular, genitivo	world's
NN+BEZ	sustantivo común, singular, contracción con verbo "to be", presente, tercera persona singular	name's
NN+HVD	sustantivo común, singular, contracción con verbo "to have", pasado	Pa'd
NN+HVZ	sustantivo común, singular, contracción con verbo "to have", presente, tercera persona de singular	summer's
NN+IN	sustantivo común, singular, contracción con preposición	buncha
NN+MD	sustantivo común, singular, contracción con auxiliar modal	sun'll
NN+NN	sustantivo común, singular, par de sustantivos unidos por guión	stomach-belly
NNS	sustantivo común, plural	members
NNS\$	sustantivo común, plural, genitivo	children's
NNS+MD	sustantivo común, plural, contracción con auxiliar modal	duds'd
NP	nombre propio, singular	Ivan
NP\$	nombre propio, singular, genitivo	Mickey's
NP+BEZ	nombre propio, singular, contracción con verbo "to be", presente, tercera persona de singular	Mack's

<b>Sustantivos</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>NP+HVZ</b>	nombre propio, singular, contracción con verbo "to have", presente, tercera persona de singular	Bill's
<b>NP+MD</b>	nombre propio, singular, contracción con auxiliar modal	Gyp'll
<b>NPS</b>	nombre propio, plural	Franciscans
<b>NPS\$</b>	nombre propio, plural, genitivo	Republicans'
<b>NR</b>	sustantivo adverbial, singular	Friday
<b>NR\$</b>	sustantivo adverbial, singular, genitivo	Saturday's
<b>NR+MD</b>	sustantivo adverbial, singular, contracción con auxiliar modal	today'll
<b>NRS</b>	sustantivo adverbial, plural	Sundays

**Tabla 7: Etiquetas de Sustantivos**

## 7. Preposiciones (Prepositions)

<b>Preposiciones</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>IN</b>	preposición	in
<b>IN+IN</b>	preposición, par de preposiciones unidas por guión	f'ovuh
<b>IN+PPO</b>	preposición, contracción con pronombre personal acusativo	t'hi-im

**Tabla 8: Etiquetas de Preposiciones**

## 8. Pronombres (Pronouns)

Pronombres		
Etiqueta	Descripción	Ejemplo
<b>PN</b>	pronombre nominal	something
<b>PN\$</b>	pronombre nominal, genitivo	one's
<b>PN+BEZ</b>	pronombre nominal , contracción con verbo "to be", presente, tercera persona de singular	nobody's
<b>PN+HVD</b>	pronombre nominal , contracción con verbo "to have", pasado	nobody'd
<b>PN+HVZ</b>	pronombre nominal , contracción con verbo "to have", presente, tercera persona de singular	somebody's
<b>PN+MD</b>	pronombre nominal , contracción con auxiliar modal	anybody'd
<b>PP\$</b>	pronombre posesivo singular	our
<b>PP\$\$</b>	pronombre posesivo plural	ours
<b>PPL</b>	pronombre reflexivo, singular	itself
<b>PPLS</b>	pronombre reflexivo, plural	ourselves
<b>PPO</b>	pronombre personal, acusativo	us
<b>PPS</b>	pronombre personal, nominativo, tercera persona de singular	she
<b>PPS+BEZ</b>	pronombre personal, nominativo, tercera persona de singular, contracción con verbo "to be", presente, tercera persona de singular	it's
<b>PPS+HVD</b>	pronombre personal, nominativo, tercera persona de singular, contracción con verbo "to have", pasado	he'd

<b>Pronombres</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>PPS+HVZ</b>	pronombre personal, nominativo, tercera persona de singular, contracción con verbo "to have", presente, tercera persona de singular	she's
<b>PPS+MD</b>	pronombre personal, nominativo, tercera persona de singular, contracción con auxiliar modal	he'll
<b>PPSS</b>	pronombre personal, nominativo, excepto tercera persona de singular	I
<b>PPSS+BEM</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con verbo "to be", presente, primera persona de singular	I'm
<b>PPSS+BER</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con verbo "to be", presente, segunda persona de singular o cualquier persona de plural	we're
<b>PPSS+BEZ</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con verbo "to be", presente, tercera persona de singular	you's
<b>PPSS+BEZ*</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con verbo "to be", presente, tercera persona de singular, negada	'tain't
<b>PPSS+HV</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con verbo "to have", presente	I've



<b>Pronombres</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>PPSS+HVD</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con verbo "to have", pasado	we'd
<b>PPSS+MD</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con auxiliar modal	you'll
<b>PPSS+VB</b>	pronombre personal, nominativo, excepto tercera persona de singular, contracción con verbo no auxiliar, presente	y'know
<b>WP\$</b>	WH-pronombre, genitivo	whosever
<b>WPO</b>	WH-pronombre, acusativo	whom
<b>WPS</b>	WH-pronombre, nominativo	that
<b>WPS+BEZ</b>	WH-pronombre, nominativo	that's who's
<b>WPS+HVD</b>	WH- pronombre, nominativo, contracción con verbo "to have", pasado	who'd
<b>WPS+HVZ</b>	WH- pronombre, nominativo, contracción con verbo "to have", presente, tercera persona de singular	that's
<b>WPS+MD</b>	WH- pronombre, nominativo, contracción con auxiliar modal	that'll

**Tabla 9: Etiquetas de Pronombres**

## **9. Cualificadores (Qualifiers)**

<b>Cualificadores</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>ABL</b>	cualificador	quite
<b>QL</b>	pre-cualificador	less

Cualificadores		
Etiqueta	Descripción	Ejemplo
QLP	post-cualificador	enough
WQL	WH-cualificador	how

Tabla 10: Etiquetas de Cualificadores

## 10. Cuantificadores (Quantifiers)

Cuantificadores		
Etiqueta	Descripción	Ejemplo
ABN	cuantificador	all
ABX	pre-cuantificador	both

Tabla 11: Etiquetas de Cuantificadores

## 11. Verbos (Verbs)

Verbos		
Etiqueta	Descripción	Ejemplo
VB	verbo, presente, imperativo o infinitivo	investigate
VB+AT	verbo presente o infinitivo, contracción con artículo	wanna
VB+IN	verbo, presente, imperativo o infinitivo contracción con preposición	lookit
VB+JJ	verbo, presente, imperativo o infinitivo contracción con adjetivo	die-dead
VB+PPO	verbo, presente, contracción con pronombre personal acusativo	gimme
VB+RP	verbo, imperativo, contracción con particular adverbial	c'mon

<b>Verbos</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>VB+TO</b>	verbo, presente, imperativo o infinitivo contracción con “to”	wanna
<b>VB+VB</b>	verbo, presente, imperativo o infinitivo, par de verbos unidos por guión	say-speak
<b>VBD</b>	verbo, pasado	produced
<b>VBG</b>	verbo, participio presente o gerundio	modernizing
<b>VBG+TO</b>	verbo, participio presente contracción con “to”	gonna
<b>VDN</b>	verbo, participio pasado	printed
<b>VDN+TO</b>	verbo, participio pasado contracción con “to”	gotta
<b>VBZ</b>	verbo, presente, tercera persona de singular	goes

**Tabla 12: Etiquetas de Verbos**

## 12. Préstamos (Foreign Words)

<b>Préstamos</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>FW-*</b>	préstamo, negación	non
<b>FW-AT</b>	préstamo, determinante artículo	la
<b>FW-AT+NN</b>	préstamo, contracción de determinante articulo y sustantivo común singular	l'orchestre
<b>FW-AT+NP</b>	préstamo, contracción de determinante articulo y sustantivo propio singular	L'Imperiale
<b>FW-BE</b>	préstamo, verbo "to be", infinitivo o imperativo	sit

<b>Préstamos</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>FW-BER</b>	préstamo, verbo "to be", presente, segunda persona de singular o cualquier persona de plural	sind
<b>FW-BEZ</b>	préstamo, verbo "to be", presente, tercera persona de singular	est
<b>FW-CC</b>	préstamo, conjunción coordinada	et
<b>FW-CD</b>	préstamo, determinante numeral cardinal	une
<b>FW-CS</b>	préstamo, conjunción subordinada	quam
<b>FW-DT</b>	préstamo, determinante pronombre, singular	hoc
<b>FW-DT+BEZ</b>	préstamo, contracción de determinante con verbo "to be", presente, tercera persona de singular	c'est
<b>FW-DTS</b>	préstamo, determinante pronombre, plural	haec
<b>FW-HV</b>	préstamo, verbo "to have", presente, excepto tercera persona de singular	habe
<b>FW-IN</b>	préstamo, preposición	de
<b>FW-IN+AT</b>	préstamo, contracción de preposición con determinante artículo	del
<b>FW-IN+NN</b>	préstamo, contracción de preposición y sustantivo común singular	d'art
<b>FW-IN+NP</b>	préstamo, contracción de preposición y sustantivo propio singular	d'Eiffel

<b>Préstamos</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>FW-JJ</b>	préstamo, adjetivo	publique
<b>FW-JJR</b>	préstamo, adjetivo, comparativo	fortiori
<b>FW-JJT</b>	préstamo, adjetivo, superlativo	óptimo
<b>FW-NN</b>	préstamo, sustantivo común singular	mano
<b>FW-NN\$</b>	préstamo, sustantivo común singular, genitivo	patronne's
<b>FW-NNS</b>	préstamo, sustantivo común plural	culpas
<b>FW-NP</b>	préstamo, sustantivo propio singular	Spagna
<b>FW-NPS</b>	préstamo, sustantivo propio plural	Atlantes
<b>FW-NR</b>	préstamo, sustantivo adverbial singular	hoy
<b>FW-OD</b>	préstamo, determinante numeral, ordinal	quintus
<b>FW-PN</b>	préstamo, pronombre personal	hoc
<b>FW-PP\$</b>	préstamo, determinante posesivo	mea
<b>FW-PPL</b>	préstamo, pronombre reflexivo singular	se
<b>FW-PPL+VBZ</b>	préstamo, contracción de pronombre reflexivo singular con verbo, presente, tercera persona de singular	s'accuse
<b>FW-PPO</b>	préstamo, pronombre personal acusativo	moi
<b>FW-PPO+IN</b>	préstamo, contracción de pronombre personal acusativo con preposición	mecum

<b>Préstamos</b>		
<b>Etiqueta</b>	<b>Descripción</b>	<b>Ejemplo</b>
<b>FW-PPS</b>	préstamo, pronombre personal, nominativo, tercera persona de singular	il
<b>FW-PPSS</b>	préstamo, pronombre personal, nominativo, excepto tercera persona de singular	ich
<b>FW-PPSS+HV</b>	préstamo, contracción de pronombre personal, nominativo, excepto tercera persona de singular con verbo "to have", presente, excepto tercera persona de singular	j'ai
<b>FW-QL</b>	préstamo, cualificador	minus
<b>FW-RB</b>	préstamo, adverbio	bas
<b>FW-RB+CC</b>	préstamo, contracción de adverbio con conjunción coordinada	forisque
<b>FW-TO+VB</b>	préstamo, contracción de "to" con verbo en infinitivo	d'entretenir
<b>FW-VB</b>	préstamo, verbo, presente, excepto tercera persona de singular, imperativo o infinitivo	esse
<b>FW-VBD</b>	préstamo, verbo, pasado	stabat
<b>FW-VBG</b>	préstamo, verbo, participio presente o gerundio	volens
<b>FW-VBN</b>	préstamo, verbo, participio pasado	verboten
<b>FW-VBZ</b>	préstamo, verbo, presente, tercera persona de singular	sigue
<b>FW-WDT</b>	préstamo, WH-determinante	que
<b>FW-WPO</b>	préstamo, WH-pronombre acusativo	quibusdam

Préstamos		
Etiqueta	Descripción	Ejemplo
FW-WPS	préstamo, WH-pronombre nominativo	qui

Tabla 13: Etiquetas de Préstamos

Es importante comentar en este punto, que los préstamos han sido tratados como una categoría única en los experimentos que se hacían sobre la totalidad del corpus, dada la gran diferencia que existen entre las palabras de distintos idiomas. Por otro lado, se han realizado también experimentos tomando únicamente las palabras extranjeras en los que sí se tiene en cuenta la categoría gramatical a la que pertenecen.

### 13. Símbolos (Symbols)

Símbolos		
Etiqueta	Descripción	Ejemplo
(	símbolo, apertura de paréntesis	(
)	símbolo, cierre de paréntesis	)
,	símbolo, coma	,
--	símbolo, guión	--
.	símbolo, finalizador de frase	. ? ; ! :
:	símbolo, dos puntos	:

Tabla 14: Etiquetas de Símbolos

Además, en este corpus se incluyen etiquetas que no aportan riqueza morfológica, sino únicamente informativa, por lo que se han obviado a lo largo de los experimentos, y van unidas por guión a las ya anteriormente mencionadas en las tablas. La etiqueta “**HL**” denota que es una palabra que aparece en un titular. La etiqueta “**TL**” palabras expresa que la palabra se encuentra en un título. La etiqueta “**NC**” significa que una palabra en negrita.

## 2.6 Desambiguación automática

Siguiendo el sentido estricto de la palabra, la desambiguación no es más que la resolución de ambigüedades, esto es, dirimir el conflicto que tiene lugar cuando un término está relacionado con dos o más temas diferentes. En este caso, la ambigüedad se produce a la hora de etiquetar algunas palabras sin tener en cuenta su contexto. Para solventar este problema, se han propuesto diversas técnicas en las que los investigadores han intentado progresar en su solución.

Las técnicas investigadas recorren desde métodos basados en diccionarios que utilizan el conocimiento codificado en recursos léxicos, a procesos de aprendizaje automático supervisado en el que un clasificador es entrenado para cada palabra distinta en un corpus de ejemplos manualmente etiquetados, pasando por métodos completamente no supervisados que agrupan las apariciones de palabras, añadiendo con ello los significados de la palabra. Entre ellos, los métodos de aprendizaje supervisado han sido los algoritmos con mayor éxito hasta la fecha.

El algoritmo de Lesk [Michael E. Lesk, 1986] es uno de los métodos basado en el diccionario más utilizado. Se basa en la hipótesis de que las palabras utilizadas juntas en el texto están relacionadas entre sí y que la relación se puede observar en las definiciones de las palabras y sus sentidos. Dos o más palabras son desambiguadas encontrando el par de sentidos del diccionario con la mayor superposición en sus definiciones. Una alternativa a la utilización de las definiciones es considerar en general la relación palabra - sentido y para calcular la similitud semántica de cada par de sentidos de palabras sobre una base de conocimiento léxico dado.

Los métodos supervisados se basan en la suposición de que el contexto puede proporcionar evidencia suficiente por sí sola para eliminar la ambigüedad de las palabras, por lo que, el sentido común y el razonamiento se considera innecesario. Los algoritmos de aprendizaje automático se han ido aplicando a la desambiguación del lenguaje natural, incluyendo técnicas tales como la selección de características, la optimización de parámetros, y el aprendizaje conjunto. Máquinas de vectores de soporte y el aprendizaje basado en memoria han demostrado los planteamientos más exitosos hasta la fecha, probablemente debido a que pueden hacer frente a la alta dimensionalidad del espacio de características. Sin embargo, estos métodos supervisados para una nueva adquisición de conocimientos suponen un cuello de botella, ya que dependen de grandes cantidades de corpus etiquetados manualmente, que son laboriosos y costosos de crear.

El proceso de aprendizaje no supervisado es el mayor desafío para los investigadores en la desambiguación lingüística. El principal supuesto es que los sentidos similares ocurren en contextos similares, y por lo tanto los sentidos pueden ser inducidos a partir del texto agrupando las ocurrencias de palabras usando alguna medida de similitud de contexto, una tarea conocida como inducción del sentido de las palabras o la discriminación. Entonces, las nuevas apariciones de la palabra se pueden clasificar en los sentidos inducidos más cercanos. El rendimiento ha sido menor que otros métodos, pero las comparaciones son difíciles, ya que los sentidos inducidos deben asignarse a un diccionario conociendo los sentidos de las palabras. Por otra parte, los métodos de inducción del sentido de las palabras pueden ser probados y comparados mediante una



aplicación. Por ejemplo, se ha demostrado que la inducción del sentido de las palabras mejora el resultado de la búsqueda Web agrupando el aumento de la calidad de los grupos de resultados y el grado de diversificación de las listas de resultados. Se espera que con el paso del tiempo y las sucesivas investigaciones, el aprendizaje no supervisado venza a la adquisición de conocimientos porque no son dependientes de esfuerzo manual.

El uso de herramientas informáticas ha dado lugar a importantes avances en el estudio de corpus lenguaje. Aunque ahora es relativamente fácil etiquetar léxicamente cada palabra en un corpus lenguaje, todavía se tiene que elegir entre numerosas formas ambiguas, especialmente en idiomas como el francés o el inglés, donde más del 70 % de las palabras son ambiguas. La lingüística computacional puede ahora proporcionar una desambiguación totalmente automática de etiquetas léxicas gracias a los etiquetados POST en poco tiempo.

## **2.7 Analizadores POST**

Como se ha demostrado anteriormente, aunque es técnicamente posible describir todos los contextos sintácticos a mano, es una tarea que se puede extender en el tiempo. Así pues, se han implementado procedimientos que pueden entrenar POS-taggers de manera automática. Estos procedimientos se pueden clasificar en dos formas diferentes: supervisados frente a no supervisados, y los basados en reglas frente a los estocásticos.

Por otra parte, también se pueden clasificar por los algoritmos reales utilizados en las implementaciones de dichos etiquetadores, ya que algunos algoritmos basados en reglas también pueden ser también estocásticos y los no supervisados se puede combinar con el entrenamiento supervisado. Los principales algoritmos utilizados son los siguientes:

- Entrenamientos manuales, no estocásticos y basado en reglas [Chanod y Tapanainen, 1995].
- No estocástico, supervisado, computación automática de reglas [Brill, 1995].
- Matrices de precedencia, bigramas o trigramas, Modelo Estándar Markov, estocástico, supervisado, con cálculo automático de matrices [Fluhr, 1977]; y posteriormente revisado [Church, 1988].
- Basado en reglas, estocástico, con reglas binarias o ternarias [Andreewsky y Fluhr, 1973]; y posteriormente revisado [Andreewsky, Debili y Fluhr, 1980].
- Modelos Ocultos de Markov, estocástico, supervisado (y sin supervisión), con el léxico conocido de antemano [Corte, Kupiec, Pedersen, y Sibún, 1992]; ampliado [Merialdo, 1994] y [Chanod y Tapanainen, 1995].

- Totalmente sin supervisión [Schütze, 1995].
- Red neuronal supervisada [Nakamura, Maruyama, Kawabata, y Shikano, 1990], [Schmid, 1994].

El principio más común para la supervisión es contar con corpus etiquetados con anterioridad (es decir, el mismo tipo de datos que un etiquetador POS produce después del entrenamiento). Si no hay ningún corpus preetiquetado disponible, es necesario construir uno desde cero. Para evitar tener que etiquetar manualmente varias decenas de miles de palabras, se puede implementar un proceso semiautomático. Un corpus corto se etiqueta por primera vez de forma manual. Este corpus se utiliza para entrenar el etiquetador POS. Entonces, un corpus más grande se etiqueta automáticamente utilizando los datos de entrenamiento del pequeño. Este corpus más grande se corrige manualmente y se utiliza para la reconversión del etiquetador. Tal proceso iterativo conducirá a grandes corpus etiquetados hasta alcanzar el tamaño necesario para la correcta formación del etiquetador.

Este procedimiento es muy útil para el entrenamiento del etiquetador en un nuevo idioma y para hacer frente a un nuevo tipo de datos lingüísticos (por ejemplo, los datos de los niños), porque pocos corpus preetiquetados están disponibles para el lenguaje de los niños. Los tamaños de los corpus de entrenamiento varían. El tamaño mínimo de las implementaciones de estos últimos es de 50.000 palabras, pero antiguos POS solían trabajar correctamente con sólo 5.000 palabras.

La variabilidad del corpus de entrenamiento es un factor importante en la reducción de la cantidad de entrenamiento necesaria. En teoría, cuanto más largo sea el conjunto de entrenamiento, mejores serán los resultados. En la práctica, hay una meseta en la mejora del etiquetado. Cuando se añade nuevo material de entrenamiento, la calidad del etiquetado mejora, y sin embargo, después de una cierta cantidad de entrenamiento, alcanza un máximo y comienza a disminuir. Esto es debido al ruido y los pequeños errores en el material de entrenamiento. La mejor cantidad de entrenamiento debe ser elegida caso por caso, ya que depende del tipo de algoritmo utilizado y la calidad del corpus de entrenamiento. Esto también explica por qué con los métodos teóricamente más poderosos no siempre se obtienen los mejores resultados.

El etiquetado no supervisado puede ser utilizado sólo por ciertas implementaciones específicas. En algunos tipos de taggers (por ejemplo, los modelos ocultos de Markov, HMM), no es necesario disponer de un corpus preetiquetado como entrada para el proceso de entrenamiento. Solo se necesita la lista de posibles categorías sintácticas para cada entrada. Desafortunadamente, este principio requiere un ajuste demasiado fino de los pesos iniciales en el modelo, y el uso de los datos tan supervisados como sea posible mejora de los resultados [Merialdo, 1994]. También es posible diseñar mecanismos automáticos para grupos de palabras en términos que se comportan del mismo modo y utilizar estos grupos para entrenar etiquetadores POS [Schütze, 1995, 1997]. Este método está reservado para corpus muy grandes y todavía no obtiene muy buenos resultados, aunque puede desarrollarse en gran medida en el futuro.

## 2.8 Herramienta de minería de datos: Aplicación Weka

La minería de datos es un área de estudio que surge de la convergencia de diversas disciplinas, como Ciencias de la Computación, Estadística, Inteligencia Artificial, Tecnología de Bases de Datos y Reconocimiento de Patrones, entre otras. Se puede definir como el proceso de análisis de datos desde diferentes perspectivas y resumirlos en información útil, que puede ser utilizada para aumentar los ingresos, reduce los costes, o ambos. Las aplicaciones de minería de datos son una de una serie de herramientas analíticas que permite a los usuarios analizar datos desde múltiples dimensiones o ángulos diferentes, categorizar, y resumir las relaciones identificadas. Técnicamente, la minería de datos es el proceso de encontrar correlaciones o patrones entre campos en grandes bases de datos relacionales.

Aunque la minería de datos es un término relativamente nuevo, la tecnología no lo es. Las compañías han utilizado potentes ordenadores para filtrar grandes volúmenes de datos y analizar los informes de investigación de mercado durante años. Sin embargo, las continuas innovaciones en la capacidad de procesamiento, almacenamiento en disco y el software de estadística, están aumentando drásticamente la precisión del análisis al tiempo que reduce el coste.

La mayoría de las herramientas de minería de datos se pueden clasificar en una de estas tres categorías: herramientas de minería de datos tradicionales, cuadro de mandos y herramientas de minería de texto. A continuación se muestra una descripción de cada uno:

- **Herramientas de minería de datos tradicionales.** Ayudan a las compañías a establecer patrones de datos y tendencias mediante el uso de una serie de algoritmos y técnicas complejas. Algunas de estas herramientas se instalan en el escritorio para monitorizar los datos y poner en relieve las tendencias, y otros capturan la información que reside fuera de la base de datos. La mayoría están disponibles en versiones Windows y UNIX, aunque algunos se especializan en un solo sistema operativo. Además, mientras que algunos pueden concentrarse en un tipo de base de datos, la mayoría son capaces de manejar los datos utilizando el procesamiento analítico online o tecnologías similares.
- **Cuadros de Mando.** Instaladas en las computadoras para controlar la información en una base de datos, los cuadros de mando reflejan los cambios y actualizaciones de datos en pantalla, a menudo en la forma de un gráfico o una tabla, que permite al usuario ver cómo se está realizando el negocio. También se puede hacer referencia a los datos históricos, lo que permite al usuario ver donde han cambiado las cosas (por ejemplo, el aumento en las ventas del mismo periodo del año anterior). Esta funcionalidad hace que los cuadros de mando sean fáciles de usar y especialmente atractivos para los administradores que deseen tener una visión general de la actuación de la empresa.

- **Herramientas de minería de texto.** El tercer tipo de herramienta de minería de datos suele encuadrarse como herramienta de minería de texto debido a su capacidad de extraer datos de diferentes tipos de texto, como por ejemplo documentos PDF, Word y/o archivos de texto plano. Estas herramientas analizan el contenido y convierten los datos seleccionados en un formato que es compatible con la base de datos de la herramienta, proporcionando así a los usuarios una manera fácil y conveniente de acceder a los datos sin la necesidad de abrir diferentes aplicaciones. El contenido escaneado puede ser desestructurado (es decir, la información está dispersa casi al azar en todo el documento, incluyendo correos electrónicos, páginas de Internet, datos de audio y vídeo) o estructurado (es decir, el formato de los datos y el propósito es conocido, como el contenido que se encuentra en una base de datos). La captura de estas entradas puede proporcionar a las organizaciones una gran cantidad de información que puede ser explotada para descubrir tendencias, conceptos y actitudes.

Además de utilizar una herramienta de minería de datos en particular, se puede elegir entre una variedad de técnicas de minería de datos. Las técnicas más utilizadas son las redes neuronales, los árboles de decisión, y el método del vecino más próximo. Cada una de estas técnicas analiza los datos de diferentes maneras:

- **Redes neuronales.** Son modelos no lineales y predictivos que aprenden a través de la formación. Aunque son técnicas de modelado predictivo poderosas, parte de su poder viene de la facilidad de uso y despliegue. Debido a su complejidad, están mejor empleadas en situaciones donde pueden ser utilizadas y reutilizadas, como la revisión de las transacciones de tarjetas de crédito cada mes para comprobar si hay anomalías.
- **Árboles de decisión.** Son estructuras en forma de árbol que representan conjuntos de decisiones. Estas decisiones generan reglas, que luego se utilizan para clasificar los datos. Los árboles de decisión son la técnica preferida para la construcción de modelos comprensibles. Se pueden utilizar para evaluar, por ejemplo, si la organización utiliza una estrategia de marketing rentable apropiado que se basa en el valor asignado para el cliente, como son las ganancias.
- **Método del vecino más próximo.** Clasifica los registros del conjunto de datos basados en datos similares en un conjunto de datos históricos. Se puede utilizar este enfoque para definir un documento que es de su interés y pedir al sistema que busque un producto similar.

Cada uno de estos enfoques aporta diferentes ventajas y desventajas que deben tenerse en cuenta antes de su uso. Las redes neuronales, que son difíciles de implementar, requieren que la entrada y la salida resultante deben expresarse numéricamente, por lo que necesitan algún tipo de interpretación en función de la naturaleza del ejercicio de la minería de datos. La técnica de árbol de decisión es la metodología más comúnmente

utilizada, debido a que es simple y fácil de implementar. Finalmente, el método del vecino más próximo se basa más en la vinculación de elementos similares y, por lo tanto, funciona mejor para la extrapolación en lugar de consultas de predicción. Independientemente de la técnica utilizada, el valor real detrás de la minería de datos es el modelado, que es el proceso de construcción de un modelo basado en criterios especificados por el usuario a partir de datos ya capturados. Una vez que se construye un modelo, que puede ser utilizado en situaciones similares donde no se conoce una respuesta

Una buena manera de aplicar técnicas de minería de datos avanzados es tener una herramienta de minería de datos flexible e interactiva que esté totalmente integrada con una base de datos. El uso de una herramienta de este tipo implicará medidas adicionales para extraer, importar y analizar los datos. En este PFC se ha utilizado la herramienta *WEKA*, que a continuación pasa a describirse.

La denominación de la herramienta *WEKA* es el acrónimo resultante de *Waikato Environment for Knowledge Analysis*, y es entorno de experimentación desarrollado en la Universidad de Waikato, Nueva Zelanda, cuya primera implementación se produjo en 1997.

El software está desarrollado en lenguaje Java (aunque originariamente fue en C) bajo la Licencia Pública General de GNU (GPL) y contiene una interfaz gráfica de usuario para interactuar con archivos de datos y producir resultados visuales (tablas y curvas de reflexión). También cuenta con una API general, por lo que se puede incrustar *WEKA*, como cualquier otra biblioteca, en aplicaciones propias para tareas automatizadas de minería de datos en la parte del servidor.

*WEKA* soporta muchas tareas de minería de datos estándar diferentes, tales como pre-procesamiento de datos, clasificación, clustering, regresión, visualización y selección de características. Su premisa básica es utilizar una aplicación informática que puede ser entrenada para realizar las capacidades de aprendizaje de máquinas y obtener información útil en forma de tendencias y patrones.

Esta herramienta opera bajo la condición de que los datos que proporciona el usuario están disponibles como un archivo plano o relación, esto significa que cada objeto de datos se describe por un número fijo de atributos que por lo general son de un tipo específico, donde normalmente los valores son alfanuméricos o numéricos. La aplicación provee a los usuarios una herramienta para identificar la información oculta de los sistemas de bases de datos y archivos sencillos de utilizar con opciones e interfaces visuales.



Figura 4: Pantalla de inicio de Weka

Al iniciar WEKA desde la consola, aparece la pantalla inicial en la que se muestran los cuatro entornos de trabajo con los que cuenta la aplicación:

- **Explorador** (*Explorer*): El modo más utilizado, descriptivo y general. Realiza las operaciones sobre un único archivo de entrada. Es el modo utilizado en este PFC.
- **Experimentador** (*Experimenter*): En este entorno se perfeccionan los experimentos y se hacen pruebas estadísticas. Se pueden aplicar varios métodos de clasificación sobre un gran volumen de datos y contrastar sus resultados para obtener estadísticas.
- **Flujo de conocimiento** (*Knowledge Flow*): Tiene básicamente las mismas funciones que el Explorador, pero con una interfaz más amigable, en la que se puede observar como la aplicación crea internamente el experimento.
- **Cliente** (*SimpleCLI*): Aporta una consola para ejecutar comandos contra Weka.

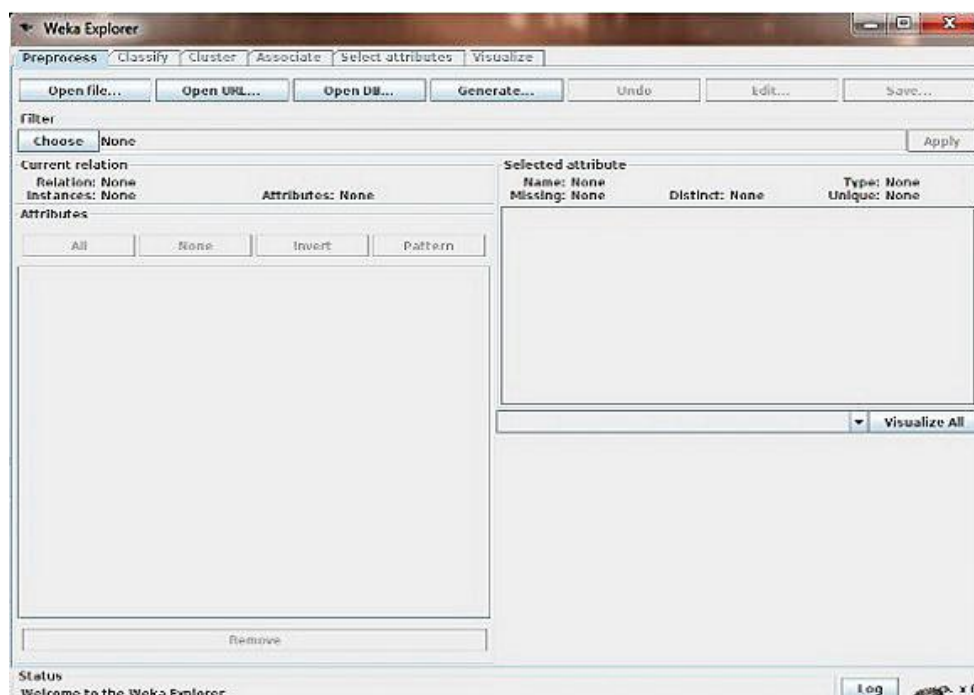


Figura 5: Ventana del modo Explorador de Weka. Parte de Preprocesado

Como se puede apreciar en la Figura 5, la pantalla del Explorador esta dividida en seis pestañas, aunque para realizar los experimentos de este PFC nos han resultado útiles sólo las dos primeras:

- **Preprocesado** (*Preprocess*): Lugar donde se define el origen de los datos de trabajo del experimento, y se pueden aplicar filtros a los mismos.
- **Clasificación** (*Classify*): En esta pantalla se pueden clasificar, bajo distintos métodos, los datos ya cargados en el paso anterior.
- **Agrupación** (*Cluster*): El funcionamiento es muy similar al de la ventana de clasificación, ya que sólo se debe elegir el método de Clustering y seleccionar las opciones que el usuario precise y pulsar en el botón *Start* para que el proceso comience.
- **Asociaciones** (*Associate*): En esta parte se aplican métodos de asociación entre datos, algo parecido a lo ya visto en las anteriores pantallas.
- **Seleccionar atributos** (*Select attributes*): Esta pestaña proporciona al usuario la posibilidad de buscar las relaciones entre los atributos que hacen que mejore los resultados.
- **Visualizar** (*Visualize*): Modo que muestra una gráfica 2D con la distribución de los atributos utilizados en el experimento, de manera que se pueden establecer correlaciones o asociaciones entre ellos.

A la hora de establecer un origen de datos desde la pestaña de *Preprocess*, Weka ofrece la posibilidad de cargar los datos desde un archivo, desde una dirección web, una base de datos o incluso generar datos adaptados a las necesidades del usuario. Los formatos admitidos en Weka son:

- **ARFF**: Es el formato por defecto que admite Weka. Internamente, se puede distinguir la parte cabecera, donde se encuentra la definición de los atributos con sus tipos, y por otra parte el cuerpo, que incluye todos los datos. La extensión del archivo es “.arff”.
- **CSV**: Documentos planos con extensión “.csv” destinados a representar tablas, donde las columnas (que en este caso representarían los atributos, que se encontrarían en la primera línea) se separan por coma (,) o punto y coma (;), y las filas por saltos de línea.
- **Codificación C4.5**: Los datos se encuentran separados en dos ficheros distintos. Uno con extensión “.names” donde se encuentran los atributos, y otro con extensión “.data” donde se hallan los datos propiamente dichos.
- **Instancias serializadas**: Objetos serializables java incluidos dentro de un archivo con extensión “.bsi”.

Tras la carga de los datos, WEKA permite revisar la información con la que se está trabajando. En la sección izquierda de la ventana del Explorador, se exponen todas las columnas de datos (atributos) y el número de filas de datos suministrados (instancias). Al seleccionar cada columna, la sección derecha de la ventana del explorador también dará información sobre los datos de esa columna del conjunto de datos. Por ejemplo, mediante la selección de una columna en la sección izquierda, la sección derecha debe cambiar para mostrar información estadística adicional sobre dicha columna. Muestra el valor máximo, el mínimo, la media y la desviación estándar (la variación esperada con respecto a la media aritmética). Por último, hay una sección con representación gráfica para examinar datos.

Para crear el modelo, se ha de pasar a la pestaña *Classify*. El primer paso es seleccionar el modelo que queremos construir, para que WEKA sepa cómo trabajar con los datos, y cómo crear el modelo adecuado. Se puede elegir entre métodos bayesianos, de árboles de decisión, funciones matemáticas, métodos de reglas e incluso meta-algoritmos (fusión de varios algoritmos).

El resultado tras aplicar el clasificador elegido, se entrenará de acuerdo con las opciones que se establezcan en el cuadro de opciones de prueba. Hay cuatro modos de prueba principales:

- **Conjunto de entrenamiento** (*Use training set*): Con esta opción Weka entrenará el método con todos los datos disponibles y luego lo aplicará otra vez sobre los mismos.
- **Conjunto de pruebas suministrado** (*Supplied test set*): Marcando esta opción, con el botón *Set...*, se podrá seleccionar un fichero de datos con el que se probará el clasificador obtenido con el método de clasificación usado y los datos iniciales.
- **Validación cruzada** (*Crossing-validation*): Se elige el número de partes en las que se quiere dividir el clasificador. Se entrenará con todas excepto con una, que servirá para probar. Este proceso se repite tantas veces como el número de partes en las que se dividió el modelo, de manera que cada vez se pruebe con una parte.
- **División porcentual** (*Percentage split*): Se define un porcentaje con el que se construirá el clasificador y con la parte restante se probará.

Para comenzar un método de clasificación se pulsará en el botón *Start*. Mientras se realiza la clasificación, la barra de estado muestra información sobre el progreso del experimento. Cuando este acabe, se mostrarán los resultados en la pantalla de salida del clasificador.



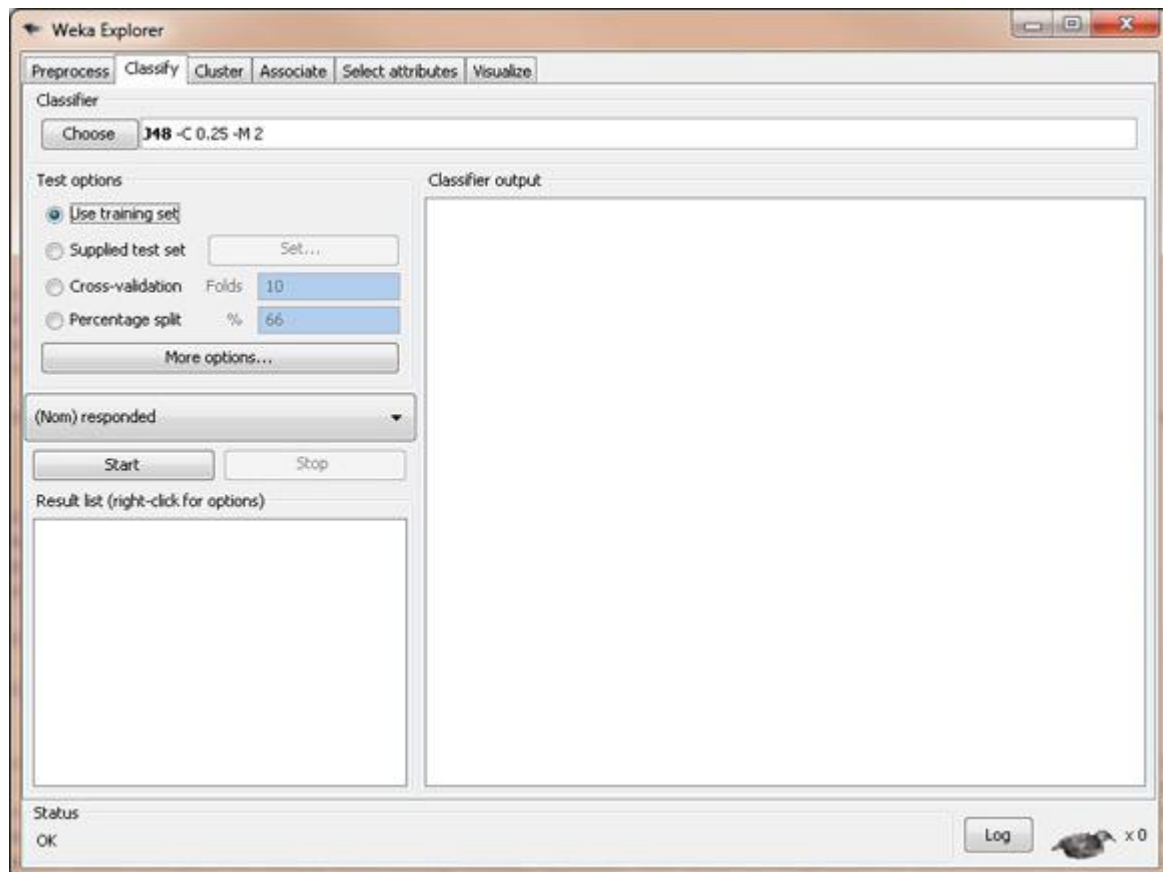


Figura 6: Pantalla del modo Explorador de Weka. Parte de Clasificación

### 2.8.1 Formato de entrada a la aplicación Weka: ARFF

Un fichero ARFF (Attribute-Relation File Format) es un archivo de texto ASCII que describe una lista de casos que comparten un conjunto de atributos. Este tipo de fichero tiene dos secciones distintas, llamadas cabecera y datos.

En la cabecera se definen el nombre de la relación, los atributos y sus tipos. Su formato es el siguiente:

@relation *nombre\_relacion*

Donde *nombre\_relacion* es un objeto String de Java.

@attribute *nombre\_atributo* *tipo\_atributo*

Donde,

*nombre\_atributo* es un objeto String de Java.

*tipo\_atributo* puede tomar el valor **numeric** (para números reales), **integer** (números enteros), **date** (fechas con formato admitido), **string** (cadenas alfanuméricas) o un conjunto de Strings definido entre llaves con sus valores separados por comas.

Por ejemplo, en este PFC, uno de los ficheros arff tiene la siguiente cabecera:

```
@relation experimentol
@attribute posicion integer
@attribute longitud integer
@attribute clase {adjective, adverb, aux, determiner,
invariant, noun, preposition, pronoun, qualifier, quantifier,
verb}
```

La sección de datos del archivo ARFF contiene la línea de declaración de datos y las líneas de instancias reales. Los datos correspondientes a cada uno de los atributos de las instancias van separados por comas. Este es un extracto de la sección de datos de un fichero ARFF utilizado en este PFC, que concuerda con los atributos definidos anteriormente:

```
@data
18582,3,noun
18638,6,noun
5872,4,verb
11523,7,pronoun
7617,2,pronoun
```

## 2.9 Máquina Virtual de Java: Oracle JRockIt

Para la ejecución de la herramienta Weka, es necesario tener instalada una máquina virtual de Java (JVM). El comando para la llamada a la aplicación es el siguiente, teniendo en cuenta que debe hacerse dentro del directorio donde se haya instalado Weka:

```
java -jar weka.jar
```

El inconveniente de dicha llamada es que el tamaño de memoria asignada a la máquina virtual será solo de 1GB, espacio que resulta muchas veces insuficiente para realizar operaciones con esta herramienta de minería de datos, con el consecuente error de desbordamiento de memoria. A priori, una solución temporal pasaría por definir un tamaño máximo asignado mayor, pero dentro de las capacidades del computador, con el modificador `-Xmx` y los megabytes elegidos.

```
java -Xmx1800m -jar weka.jar
```

Durante los experimentos realizados para este PFC, y aún utilizando la sintaxis de llamada anterior, se llegó al desbordamiento de memoria, por lo que se buscó otra máquina virtual de Java diferente a la estándar proporcionada por Oracle, y el resultado fue la máquina virtual *Oracle JRockit*.

La máquina virtual Oracle JRockit es una JVM de alto rendimiento desarrollada para garantizar la fiabilidad, escalabilidad, capacidad de administración y flexibilidad para las aplicaciones Java. Proporciona un rendimiento mejorado en aplicaciones Java desplegadas en arquitecturas tanto de 32 como de 64bits. Esta JVM es un componente de la JRockit JDK (Java Development Kit). Además de la JRockit JVM, la JDK contiene el entorno de ejecución de Java (JRE). La JRE contiene las bibliotecas de clases de Java (según lo especificado por la plataforma Java 6 Standard Edition) y un conjunto de herramientas de desarrollo, como un compilador.

La máquina virtual JRockit es capaz de compilar código just-in-time (JIT) y optimizarlo para garantizar un alto rendimiento, lo que repercute positivamente tanto en tiempo como en recursos. Igualmente, también gestiona de una mejor manera los hilos de las ejecuciones y los interbloqueos. Pero lo que realmente ha hecho de JRockit una herramienta útil en este PFC es la mejora en la gestión de memoria con respecto a la JVM estándar de Java. Tanto la asignación de memoria para objetos como la recolección de basura han sido perfeccionadas.

Los objetos Java residen en una zona llamada montículo (más conocido como *heap*, en inglés), una sección de la pila de memoria que se crea cuando la JVM se inicia. El montículo puede aumentar o disminuir, mientras que la aplicación es ejecutada. Cuando la pila está llena, actúa el recolector de basura. JRockit identifica los espacios de memoria que contienen objetos que se están utilizando (objetos vivos). A continuación, recupera los espacios de memoria que no contienen objetos vivos, convirtiéndolos en espacios disponibles para la asignación a los nuevos objetos. A veces el montículo se divide en dos partes: la parte joven y la parte vieja. La parte joven es la zona de la pila donde se crean los objetos nuevos mientras haya espacio, momento en el que entra en acción el recolector de basura, moviendo los objetos vivos a otra zona de la pila diferente, que es la parte vieja. Cuando la parte vieja está llena, se ejecuta igualmente en esa zona el recolector de basura. Al tener dos zonas diferenciadas, se consigue mayor agilidad en la administración de memoria, ya que los objetos con ciclo de vida corto dejarán pronto sitio a otros.

JRockit también distingue entre objetos pequeños (menores de 128KB) y objetos grandes. Los objetos pequeños se alojan en áreas locales de los hilos, fragmentos del montículo reservados en exclusiva para el hilo al que pertenecen sin necesidad de estar sincronizados con el resto de los hilos en ejecución. Los grandes objetos van directamente al montículo, pues requieren mayor sincronización entre los subprocesos de Java, aunque JRockit posee un sistema que gestiona fragmentos libres de diferentes tamaños para mejorar la velocidad de asignación y reducir la sincronización.

Es posible que después de una recolección de basura, el montículo se fragmente, es decir, que existan numerosos espacios libres pequeños, por lo que la asignación de objetos grandes se complicaría. Para reducir la fragmentación, JRockit compacta una parte del montículo en cada recolección de basura. La compactación mueve los objetos más hacia el fondo del montículo, creando zonas libres más grandes en la parte superior de la pila. El tamaño (y la posición) de la zona de compactación y el método de compactación son seleccionados por la heurística avanzada, dependiendo del modo de recolección de basura utilizado. La compactación se lleva a cabo al comienzo o durante la fase de barrido y mientras que los hilos de Java estén detenidos.

Para utilizar esta nueva JVM en lugar de la original de Java, basta con ejecutar el archivo `java.exe` que contiene JRockit en la carpeta “bin” de su lugar de instalación. El resto del comando es completamente idéntico.

### 3. METODOLOGÍA

El punto de partida de este PFC se encuentra en la búsqueda de un corpus en inglés, en concreto el Brown Corpus, con el que poder realizar experimentos para la desambiguación morfológica a través de textos etiquetados. El corpus encontrado resulta estar casi completamente etiquetado, por lo que termina de etiquetarse de manera manual utilizando el mismo juego de etiquetas para las palabras que no reflejan su categoría gramatical.

Se han realizado diversos experimentos utilizando una muestra del corpus, otros sobre la totalidad del corpus pero sólo clasificando las palabras con caracteres alfabéticos y por otro lado, unos en los que únicamente intervienen las palabras extranjeras.

En cada tipo de experimento, se ha dividido el proceso en dos fases. En la primera fase, el proceso de desambiguación no tiene en cuenta el contexto de las palabras. En la segunda fase, por el contrario, los experimentos parten con el mejor resultado de la primera fase como reglas de desambiguación, y además cuentan con el contexto de las palabras para intentar conseguir mayor porcentaje de acierto.

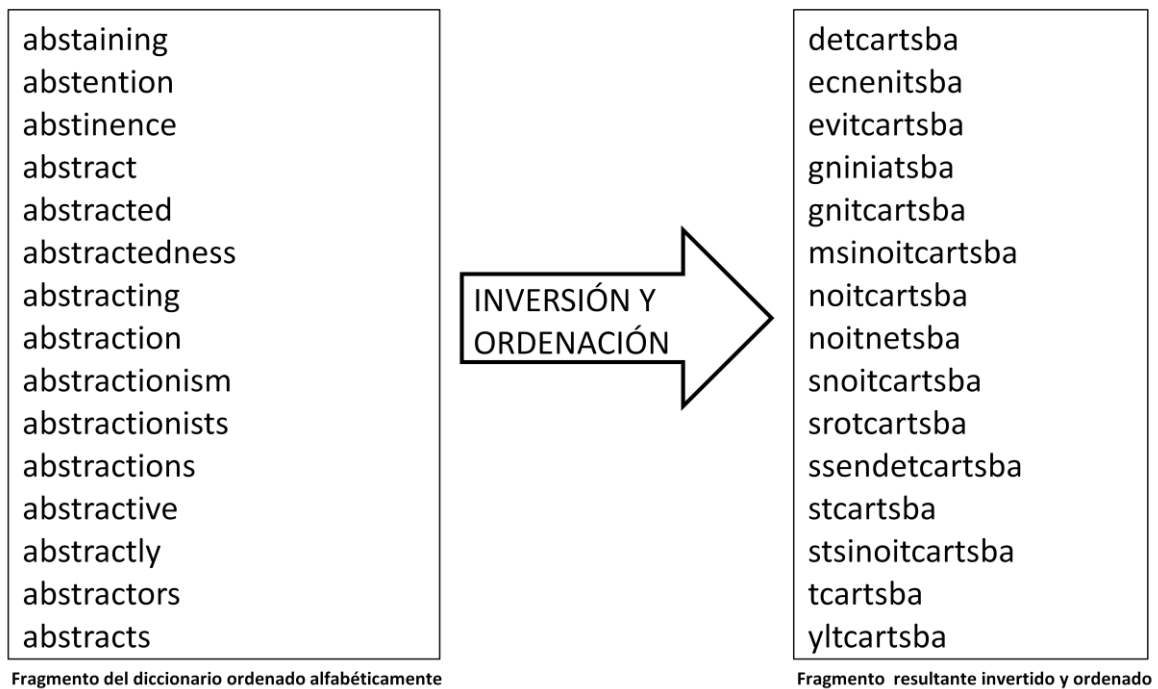
#### 3.1 Primera Fase de Metodología

En la primera fase de metodología la característica principal es que no se tiene en cuenta el contexto de las palabras. Los experimentos se han realizado sobre la totalidad del corpus, exceptuando los signos de puntuación y los vocablos procedentes de otras lenguas, pues contaminarían los resultados. El diccionario sobre el que se apoya el experimento está compuesto por todos los términos ingleses del corpus.

En esta primera fase, se generan reglas de desambiguación en función de la palabra a etiquetar, teniendo en cuenta el número de letras que tiene dicha palabra, su posición dentro del diccionario y la etiqueta que la categoriza en el corpus.

Para la realización de esta primera fase, primero se generó un diccionario de términos. El origen ideal para crearlo es el propio corpus. Una vez obtenido el diccionario, a la hora de crear el fichero de entrada a Weka, se carga la totalidad del diccionario, se invierten las palabras y se guardan en una lista ordenadas alfabéticamente. De esta forma, el diccionario queda ordenado por los finales de los términos, lo que realmente aporta más información para disipar la categoría gramatical de una palabra.

A parte de la utilización del diccionario, también se necesita un fichero de mapeo entre las etiquetas del corpus con la categoría gramatical general. De igual manera, la aplicación que se encarga de generar el fichero con extensión “.arff”, requiere un archivo de conversión de los vocablos compuestos que aparecen en el corpus, a términos simples con sus etiquetas propias. Una vez reunidos todos los archivos, el resultado del proceso arroja el fichero de entrada a Weka.



**Figura 7: Ejemplo de conversión del diccionario**

La relación que se expresará en el fichero Weka para cada una de las palabras que aparecen en el Brown Corpus será la siguiente:

posicion, longitud, clase

Donde,

- **posicion** es el lugar en el que se ubica la palabra del corpus dentro del diccionario invertido y ordenado.
- **longitud** es el número de caracteres de la palabra del corpus.
- **clase** es la categoría gramatical general de la palabra del corpus que se corresponde con su etiqueta.

```
@relation experimentol

@attribute posicion integer
@attribute longitud integer
@attribute clase {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}

@data
18582,3,noun
18638,6,noun
5872,4,verb
11523,7,pronoun
7617,2,pronoun
5886,5,aux
33856,6,verb
19958,3,noun
23692,6,noun
7617,2,pronoun
24875,3,aux
17524,4,adverb
5748,3,adjective
```

**Figura 8: Fragmento del fichero de entrada a Weka para la 1ª fase de experimentación**

A partir de las tres características anteriores, se obtienen las reglas de desambiguación de las categorías gramaticales. Dichas reglas tienen un formato condicional, en el que intervienen *posicion* y *longitud*. Por ejemplo, se pueden dar una regla en la que se exprese que si el valor de *posicion* es superior a 30 y *longitud* es menor que 6, el valor de *clase* será “*noun*”. Esto es así porque como el diccionario esta ordenado por las terminaciones, ya que las palabras que terminen igual estarán en posiciones sucesivas y agrupadas. Además, como los sufijos son la mayoría de las veces los encargados de definir la categoría gramatical, se obtiene una buena fuente de información veraz a la hora de catalogar las palabras.

### 3.2 Segunda Fase de Metodología

En la segunda fase de metodología la característica principal se fundamenta en que se tiene en cuenta el contexto de las palabras. Los experimentos se han realizado sobre la totalidad del corpus, exceptuando los signos de puntuación y los vocablos procedentes de otras lenguas, pues contaminarían los resultados. El diccionario sobre el que se apoya el experimento está compuesto por todos los términos ingleses del corpus.

En esta segunda fase entran en juego las reglas de desambiguación generadas en la primera fase procedentes del resultado de la clasificación con el meta-algoritmo AdaBoostM1 aplicado al Part.

Durante la segunda fase se quiere realizar la desambiguación atendiendo también al contexto de cada palabra. Es decir, que a la vez que se realiza el proceso de categorización sobre una palabra, se hace también sobre la anterior y la siguiente, con el objetivo de afinar más aún los resultados. He aquí un ejemplo visual del proceso de etiquetación de una frase completa:

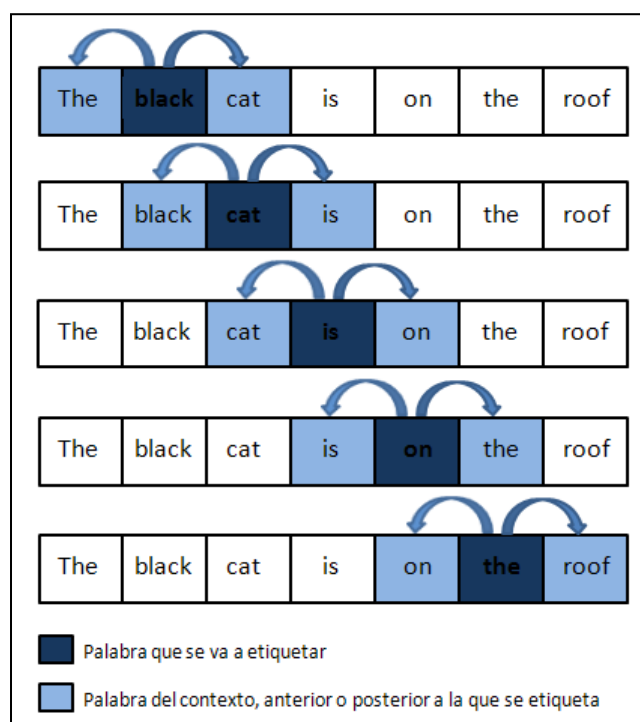


Figura 9: Proceso de etiquetado con contexto

Para la realización del fichero de entrada de la segunda fase son necesarios el fichero de entrada de la primera fase y las reglas de desambiguación generadas por algoritmo elegido. Se ha desarrollado un proceso Java que es capaz de, a partir de cada instancia generada en el fichero de la primera fase, contrastando el valor de *posición* y *longitud* con las reglas de desambiguación, devolverá la posición de la regla en la que se ha clasificado la palabra y la etiqueta que correspondería.

Se han establecido dos tipos de generación del fichero de esta segunda fase de experimentación. En uno de ellos sólo se cuenta con el número de regla correspondiente a la palabra que se quiere desambiguar, mientras que en el otro también se obtendrán los números de las reglas de las palabras del contexto. En ambos casos, siempre están presentes las categorías gramaticales correspondientes en las reglas de desambiguación tanto de la palabra cuestionada como las de su contexto. Se muestra a continuación un breve fragmento de ambos tipos de fichero:

```
@relation experimento1Fase2_1R

@attribute etiquetaactual {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}
@attribute reglaactual integer
@attribute etiquetaanterior {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}
@attribute etiquetaposterior {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}
@attribute etiquetacorpus {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}

@data
noun,653,noun,verb,noun
verb,1636,noun,pronoun,verb
pronoun,489,verb,pronoun,pronoun
pronoun,2588,pronoun,aux,pronoun
aux,1300,pronoun,noun,aux
noun,821,aux,noun,verb
```

**Figura 10: Fragmento del fichero de entrada a Weka para la 2ª fase de experimentación con sólo número de regla para la palabra a desambiguar**

```
@relation experimento1Fase2_3R

@attribute etiquetaactual {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}
@attribute reglaactual integer
@attribute etiquetaanterior {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}
@attribute reglaanterior integer
@attribute etiquetaposterior {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}
@attribute reglaposterior integer
@attribute etiquetacorpus {adjective, adverb, aux, determiner, invariant, noun,
preposition, pronoun, qualifier, quantifier, verb}

@data
noun,653,noun,3068,verb,1636,noun
verb,1636,noun,653,pronoun,489,verb
pronoun,489,verb,1636,pronoun,2588,pronoun
pronoun,2588,pronoun,489,aux,1300,pronoun
aux,1300,pronoun,2588,noun,821,aux
noun,821,aux,1300,noun,3481,verb
```

**Figura 11: Fragmento del fichero de entrada a Weka para la 2ª fase de experimentación con número de regla para la palabra y su contexto**



### 3.3 Consideraciones con el Corpus

Tras llevar a cabo los primeros experimentos de adaptación y asimilación de conocimientos, se obtuvieron ciertas conclusiones para llevar a buen puerto la consecución del proyecto. La primera de ellas es que debía utilizarse la totalidad del corpus para buscar mayor variabilidad de palabras. También se deben mapear las etiquetas en las categorías gramaticales generales para focalizar en mayor medida el acierto de la desambiguación. Además, también se cambia de máquina virtual para obtener un mayor rendimiento y fiabilidad en el consumo de recursos.

La correspondencia entre las etiquetas y la categoría general se hace durante la generación del fichero de la primera fase de experimentación, y es la siguiente:

- **Adjetivo (Adjective)**

**JJ** (adjetivo calificativo), **JJ\$** (adjetivo con genitivo), **JJR** (adjetivo comparativo), **JJS** (adjetivo semánticamente superlativo) y **JJT** (adjetivo superlativo).

- **Adverbio (Adverb)**

**\*** (adverbio de negación), **RB** (adverbio), **RB\$** (adverbio con genitivo), **RBR** (adverbio comparativo), **RBT** (adverbio superlativo), **RN** (adverbio nominal), **RP** (adverbio particular), **WRB** (adverbio interrogativo).

- **Auxiliar (Aux)**

**BE, BED, BED\*, BEDZ, BEDZ\*, BEG, BEM, BEM\*, BEN, BER, BER\*, BEZ, BEZ\*** (verbo “to be” en diversas personas y tiempos verbales, incluyendo negaciones)

**DO, DO\*, DOD, DOD\*, DOZ, DOZ\*** (verbo “to do” en diversas personas y tiempos verbales, incluyendo negaciones)

**HV, HV\*, HVD, HVD\*, HVG, HVN, HVZ, HVZ\*** (verbo “to have” en diversas personas y tiempos verbales, incluyendo negaciones)

**MD, MD\*** (verbos auxiliares modales, con y sin negaciones, respectivamente).

- **Determinante (Determiner)**

**AP** (determinante), **AP\$** (determinante con genitivo), **AT** (determinante articulo), **CD** (determinante numeral cardinal), **CD\$** (determinante numeral cardinal con genitivo), **DT** (determinante pronombre singular), **DT\$** (determinante pronombre con genitivo), **DTI** (determinante pronombre indeterminado), **DTS** (determinante pronombre plural), **DTX** (determinante pronombre), **OD** (determinante numeral ordinal), **WDT** (determinante interrogativo).

- **Invariable (Invariant)**

**CC** (conjunción coordinada), **CS** (conjunción subordinada), **TO** (palabra auxiliar “to”),

**UH** (interjección).

- **Sustantivo (Noun)**

**NN** (sustantivo común), **NN\$** (sustantivo común con genitivo), **NNS** (sustantivo común plural), **NN\$S** (sustantivo común plural con genitivo), **NP** (nombre propio singular), **NP\$** (nombre propio singular con genitivo), **NPS** (nombre propio plural), **NPS\$** (nombre propio plural con genitivo), **NR** (sustantivo adverbial singular), **NR\$** (sustantivo adverbial singular con genitivo), **NRS** (sustantivo adverbial plural).

- **Preposición (Preposition)**

**IN** (preposición).

- **Pronombre (Pronoun)**

**PN** (pronombre nominal), **PN\$** (pronombre nominal con genitivo), **PP\$** (pronombre posesivo singular), **PP\$S** (pronombre posesivo plural), **PPL** (pronombre reflexivo singular), **PPLS** (pronombre reflexivo plural), **PPO** (pronombre personal acusativo), **PPS** (pronombre personal tercera persona singular), **PPSS** (pronombre personal excepto tercera persona singular), **WP\$** (pronombre interrogativo con genitivo), **WPO** (pronombre interrogativo acusativo), **WPS** (pronombre interrogativo nominativo).

- **Cualificador (Qualifier)**

**ABL** (cualificador), **QL** (pre-cualificador), **QLP** (post-cualificador), **WQL** (cualificador interrogativo).

- **Cuantificador (Quantifier)**

**ABN** (cuantificador), **ABX** (pre-cuantificador).

- **Verbo (Verb)**

**VB** (verbo presente, imperativo o infinitivo), **VBD** (verbo pasado), **VBG** (verbo participio presente o gerundio), **VBN** (verbo participio pasado), **VBZ** (verbo presente tercera persona singular)

De la misma manera, también se han mapeado las etiquetas que representan signos de puntuación a la categoría **Símbolo (Symbol)**, pero han sido obviadas en los experimentos en pos de un resultado más veraz, dado que el reconocimiento de los símbolos es trivial y carece de interés investigador. Los préstamos, etiquetados con “FW” además de su categoría gramatical, se han mapeado con la categoría **Extranjerismo**

**(ForeignWord)** también han sido ignorados en el experimento total, pero han sido objeto de estudio exclusivo en otros experimentos aparte.

Por otra parte, durante la revisión del corpus, se detectaron numerosas palabras compuestas, las cuales contenían etiquetas compuestas, representando las categorías gramaticales de las palabras que componían la unión. Se han recopilado todas las palabras compuestas para generar un fichero de mapeo que sirva para separarlas en los vocablos únicos y sus categorías y de esa manera tratarlas de la misma forma que al resto de los tokens que componen el corpus. También con esto se pretende conseguir mayor fiabilidad en los resultados. A continuación se muestran algunos de los mapeos que se realizan a las palabras compuestas que se encuentran en Brown Corpus:

```
i'm/ppss+bem -> i/ppss + am/bem  
i'll/ppss+md -> i/ppss + will/md  
i'd/ppss+md -> i/ppss + should/md  
i'd/ppss+hvd -> i/ppss + had/hvd  
i've/ppss+hv -> i/ppss + have/hv  
you're/ppss+ber -> you/ppss + are/ber  
you'll/ppss+md -> you/ppss + will/md  
you'd/ppss+md -> you/ppss + should/md  
you'd/ppss+hvd -> you/ppss + had/hvd  
you've/ppss+hv -> you/ppss + have/hv  
you's/ppss+bez -> you/ppss + is/bez
```

**Figura 12: Frágmento del fichero de correspondencia de palabras compuestas**

## 4. DESARROLLO

### 4.1 Aplicación desarrollada: ArffGenerator

Para la creación del diccionario y de los ficheros de entrada a la aplicación de minería de datos Weka, se ha desarrollado un programa en lenguaje Java llamado “ArffGenerator”, el cual está en proceso de copyright por parte de la Universidad Carlos III y pasará a formar parte de su propiedad intelectual.

En este sistema la descripción es una descomposición en tres capas que busca minimizar las dependencias existentes entre cada una de las capas. De este modo se busca maximizar las opciones de mantenibilidad de la aplicación, así como que sea posible la sustitución de componentes de forma que el resto del sistema no se vea afectado. El sistema sigue la descomposición que se muestra en la siguiente ilustración:

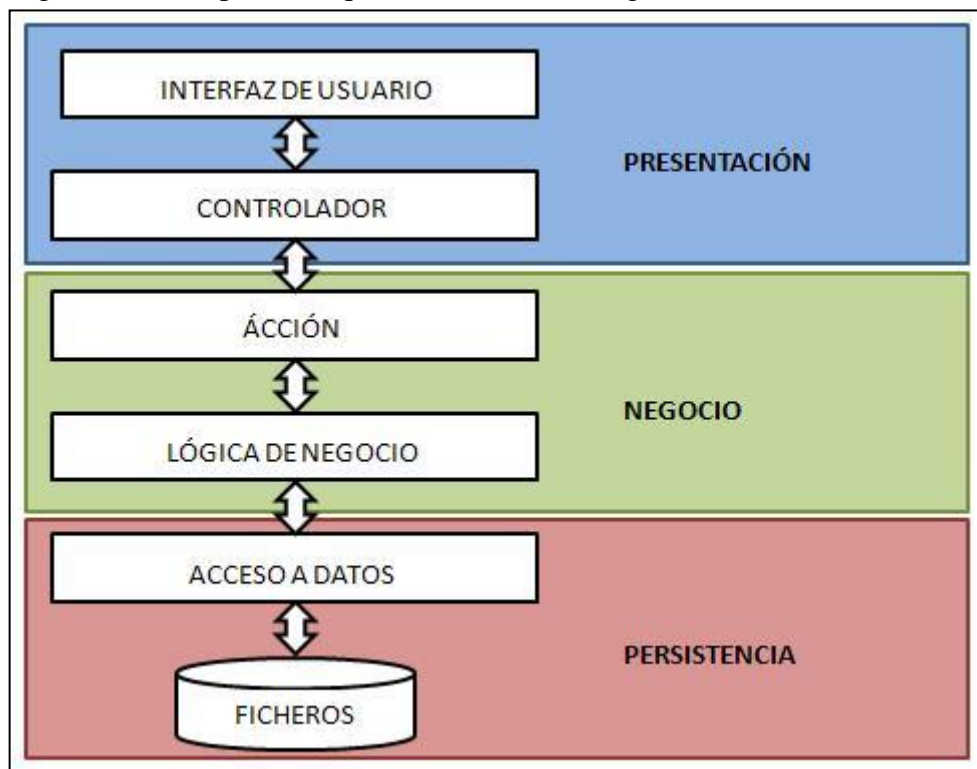


Figura 13: Esquema de arquitectura del sistema

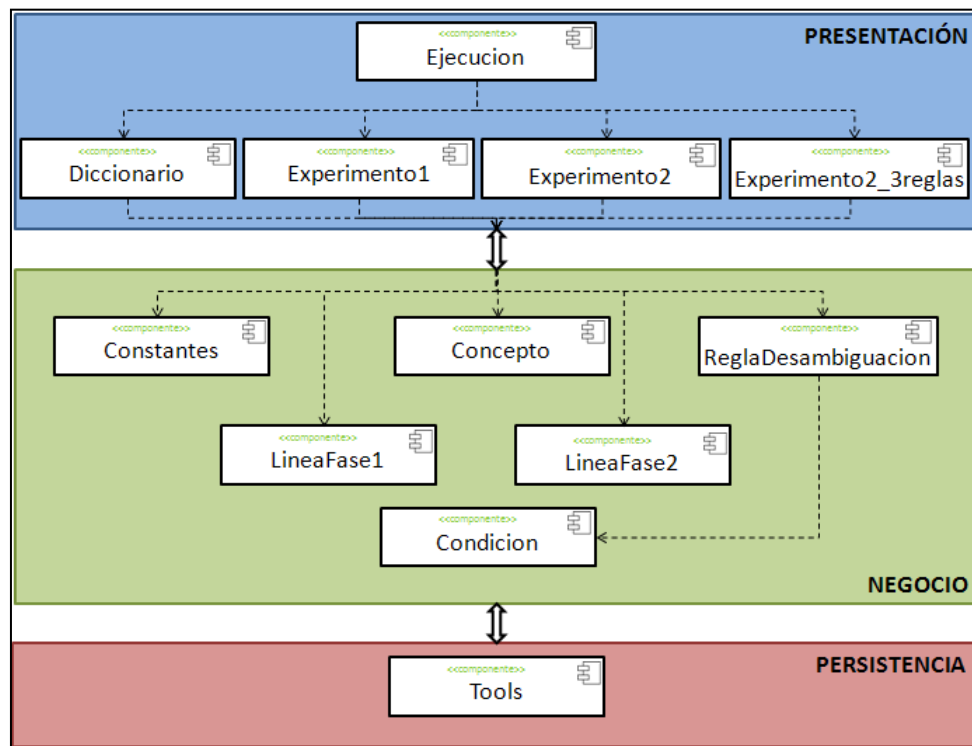
Las capas en las que se ha dividido el sistema son, como puede verse en la figura anterior, las siguientes:

**Presentación:** Los componentes que formen esta capa serán todos aquellos con los que puedan interactuar los usuarios. A través de esta capa, los usuarios podrán elegir sus opciones y verán los resultados de las mismas. El punto de unión con el resto del sistema es a través de un controlador.

**Negocio:** En esta capa reside la lógica que permite realizar las operaciones solicitadas por el usuario contra los datos almacenados. Aquí es donde se ejecutan los procesos que satisfacen las peticiones de los usuarios y se envían los resultados de los mismos a la capa de Presentación.

**Persistencia:** Lugar donde residen los datos y los gestores que controlan la manipulación de los mismos, permitiendo el acceso, recuperación y/o generación de información.

A continuación se muestra descomposición real de las capas en los componentes que las forman:

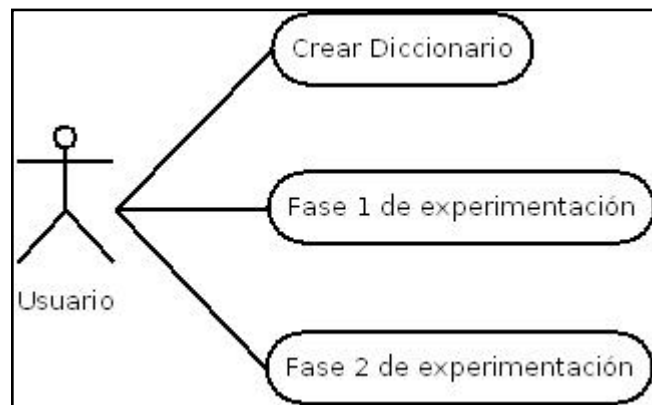


**Figura 14: Diagrama de componentes de la aplicación**

En el diagrama de casos de uso mostrado seguidamente, se define lo que hace ArffGenerator desde un punto de vista externo. Estos son los diferentes casos de uso:

- **Crear Diccionario:** Se genera un listado de palabras con todas las que aparecen en el Corpus. En este caso, se han tomado únicamente las palabras que contienen caracteres alfabéticos.
- **Primera fase de experimentación:** En la que se reúnen los datos necesarios de las palabras a clasificar para poder generar las reglas de desambiguación.

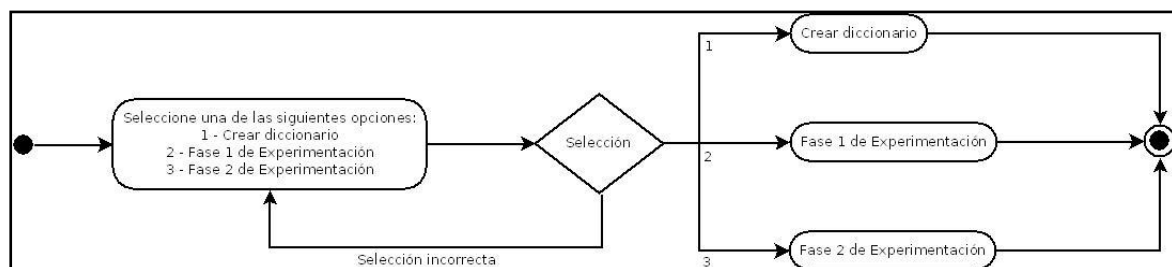
- Segunda fase de experimentación: En este caso de uso se obtienen los datos necesarios de las palabras y de su contexto para después poder generar las reglas de desambiguación.



**Figura 15: Diagrama de casos de uso**

A continuación se muestran, por separado, cada uno de los diagramas de actividad en los que se divide la ejecución de la aplicación ArffGenerator.

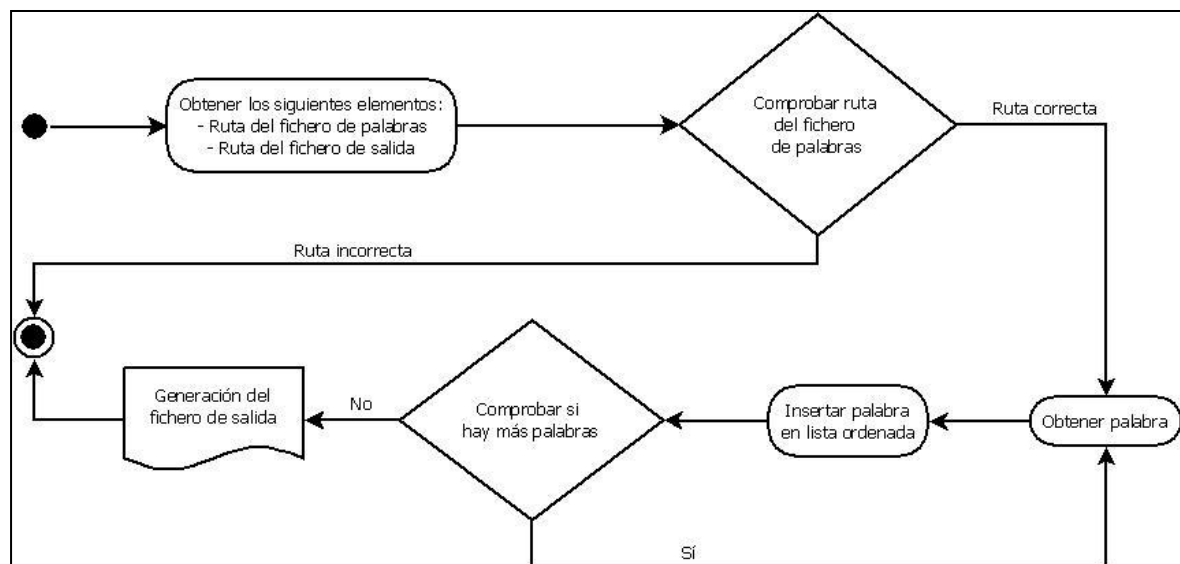
En primer lugar, en la Figura 16 se puede observar el proceso de ejecución del menú general de la aplicación, en el que se pide al usuario escoger una de las opciones. Si éste no pulsa los números 1, 2 o 3, la aplicación mostrará un mensaje avisando de la selección incorrecta, permitiendo de nuevo la selección por parte del usuario.



**Figura 16: Diagrama de actividades del Menú Principal**

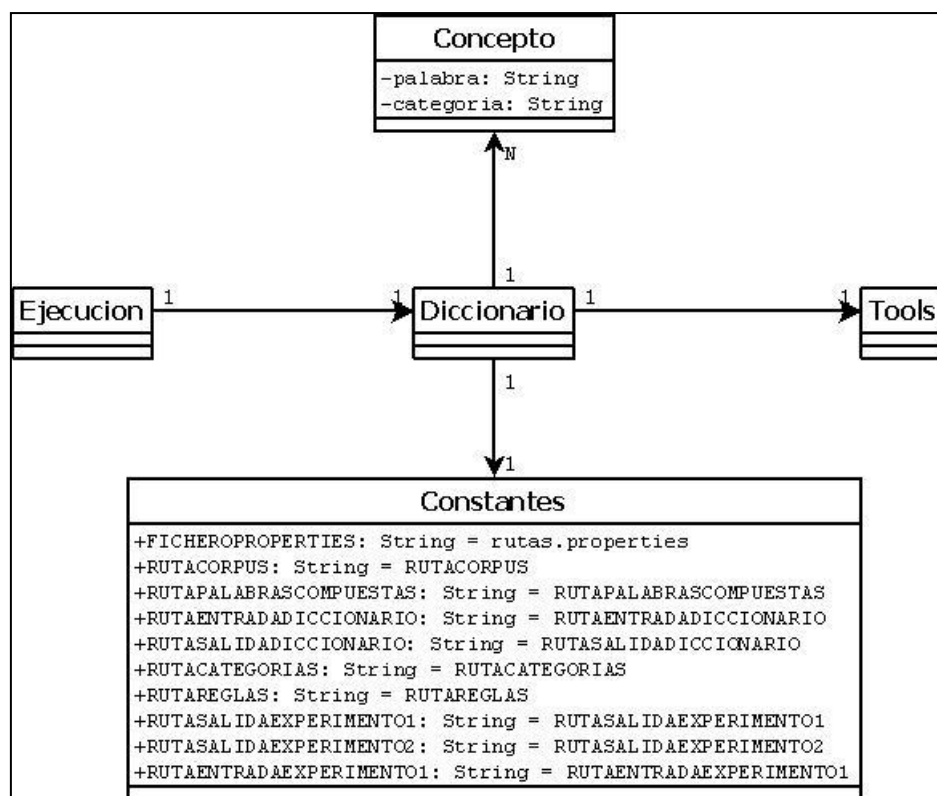
En la Figura 17 se muestra el proceso de generación del diccionario a partir de una ruta específica.

Como se puede apreciar, el primer paso consiste en comprobar que la ruta determinada sea la correcta para continuar con el proceso; en caso contrario, terminaría su ejecución advirtiéndole al usuario del error. Tras validar la existencia del fichero/directorio (este proceso es capaz de navegar entre carpetas buscando los ficheros de texto, haciendo más ágil la obtención del diccionario del corpus), se accede a un bucle en el que se irán obteniendo las palabras que conformarán el diccionario, ordenadas alfabéticamente en una lista en memoria principal y sin repetición. Al llegar al final del proceso, se vuelca el contenido de la lista generada en un fichero ubicado en la ruta especificada en el fichero de propiedades.



**Figura 17: Diagrama de actividad de la creación del Diccionario**

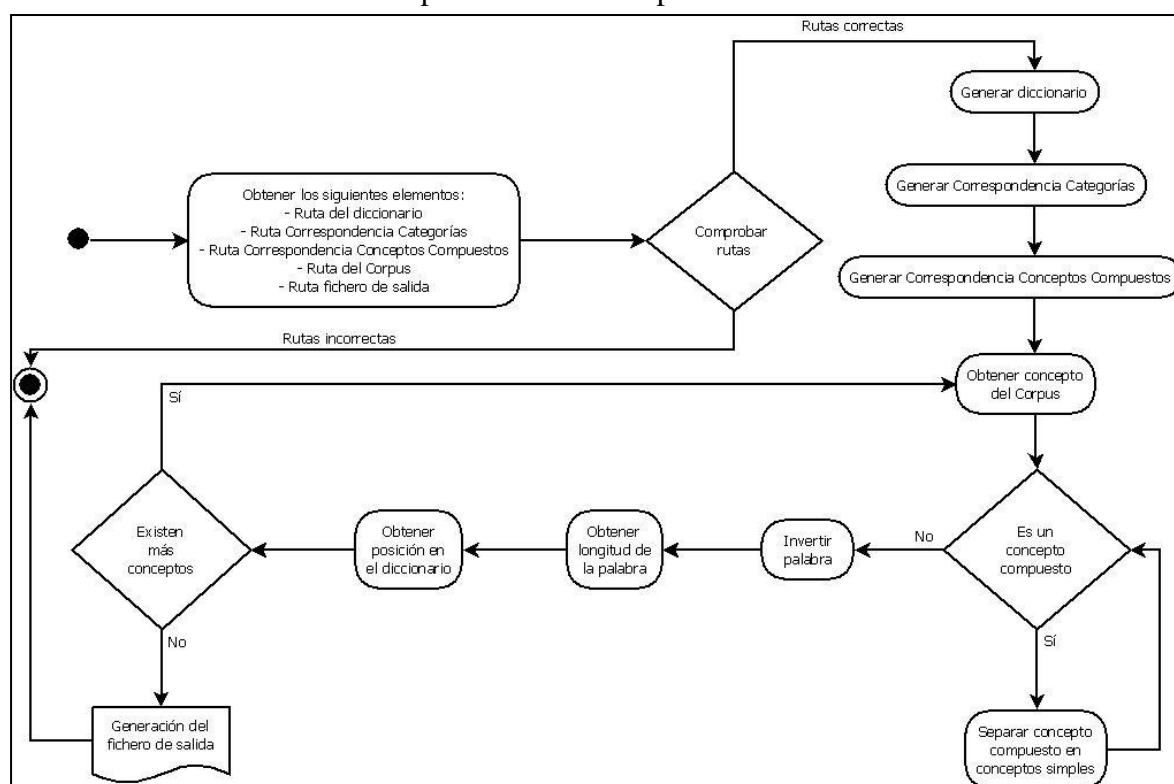
El diagrama de clases para la generación del diccionario es el siguiente:



**Figura 18: Diagrama de clases para la generación del diccionario**

El proceso de generación del fichero para la fase 1 de experimentación se muestra en la Figura 19. Para ello se necesitan como parámetros las rutas del diccionario (generado en el paso anterior, ver Figura 17), las correspondencias de las categorías y de los conceptos compuestos, del corpus y la del fichero de salida generado al finalizar la ejecución.

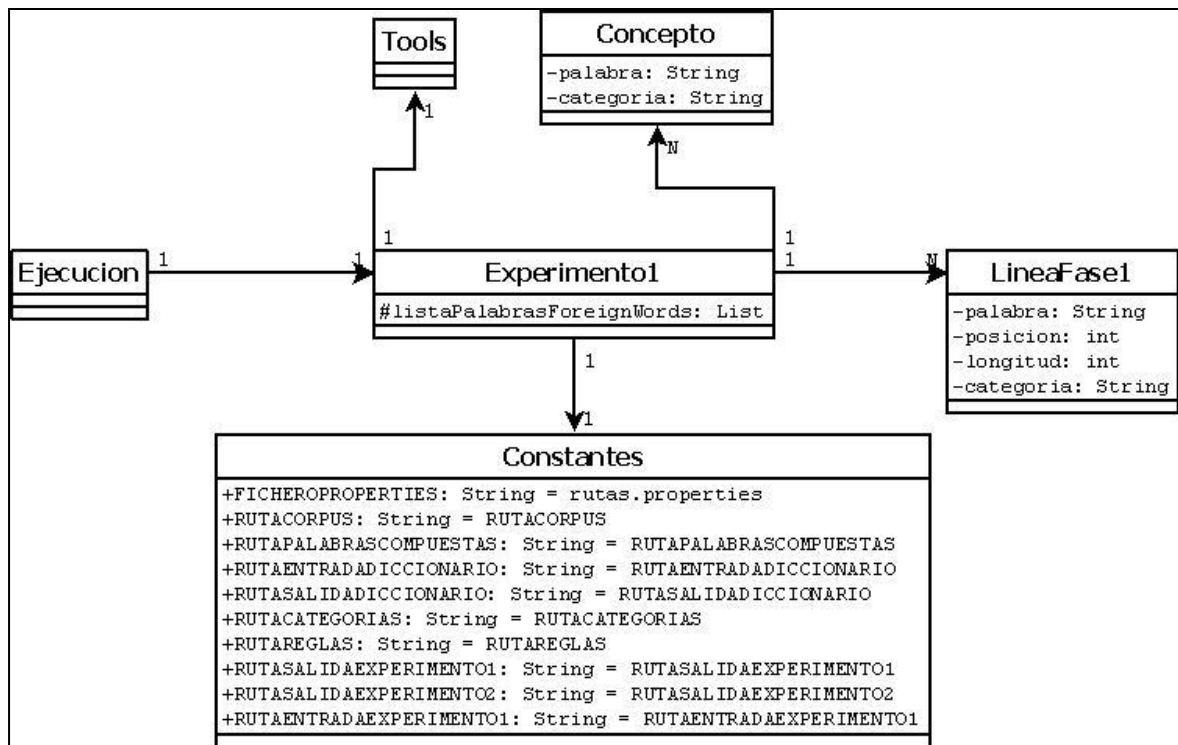
Para evitar posteriores errores, como primer paso se comprueba que las rutas designadas sean correctas o se terminaría la ejecución avisando al usuario del error cometido. Tras esta validación, se generan, sucesivamente, listas en memoria principal que contienen el diccionario (invirtiendo las palabras que lo componen y ordenando el resultado) y las correspondencias entre las categorías gramaticales y las etiquetas, y también las correspondencias de los conceptos compuestos con sus conceptos simples equivalentes. Llegados a este punto es conveniente aclarar que un concepto es la denominación que se ha adoptado en llamar al grupo formado por una palabra del corpus y su etiqueta. Posteriormente se carga también el corpus en memoria principal. Una vez realizadas todas estas cargas, se accede a un bucle en el que se irán obteniendo los conceptos que conforman el corpus, se comprobará si es un concepto compuesto (en cuyo caso se separaría en los conceptos simples que lo conformasen), se invertirá la palabra, se obtendrá su número de caracteres y también la posición que ocupa dentro del diccionario ubicado en memoria principal. Al llegar al final del proceso, se vuelca el contenido de la lista generada en un fichero ubicado en la ruta especificada al inicio para así crear el fichero de entrada a Weka de la primera fase de experimentación.



**Figura 19: Diagrama de actividad de la creación del fichero de la Fase 1 de Experimentación**

El diagrama de clases para la generación del fichero para la primera fase de experimentación es el siguiente:





**Figura 20: Diagrama de clases para la generación del fichero de la Fase 1 de experimentación**

En la Figura 21 se detalla el proceso de creación del fichero para la fase 2 de experimentación. Para ello se necesitan como parámetros las rutas del fichero de entrada de la fase 1 de experimentación (generado en el paso anterior, ver Figura 19), la del fichero que contienen las reglas correspondientes al meta-algoritmo AdaBoostM1-PART y la del fichero de salida generado al finalizar la ejecución.

Como siempre y para evitar posteriores errores, en el primer paso se comprueba que las rutas introducidas sean correctas o se terminaría la ejecución avisando al usuario del error cometido. Tras esta validación, se pregunta al usuario si desea aplicar las reglas de desambiguación a las palabras del contexto, para conformar un fichero con 5 o 7 atributos (dependiendo de la presencia o no de las reglas de desambiguación para el contexto). Seguidamente, se carga en memoria principal las reglas procedentes del resultado obtenido en Weka en la fase 1 de experimentación tras aplicar el meta-algoritmo AdaBoostM1-PART. Posteriormente, se cargan los conceptos contenidos en el fichero de la fase 1 de experimentación. Siguiendo la misma política que en se mostró en la Figura 19, se accede a un bucle en el que se irán obteniendo los conceptos que conforman el fichero, se obtiene la regla en la cual se encuadraría dicho concepto y la información del contexto. Al llegar al final del proceso, se vuelca el contenido de la lista generada en un fichero ubicado en la ruta especificada en el archivo de propiedades para así crear el fichero de entrada a Weka de la segunda fase de experimentación.

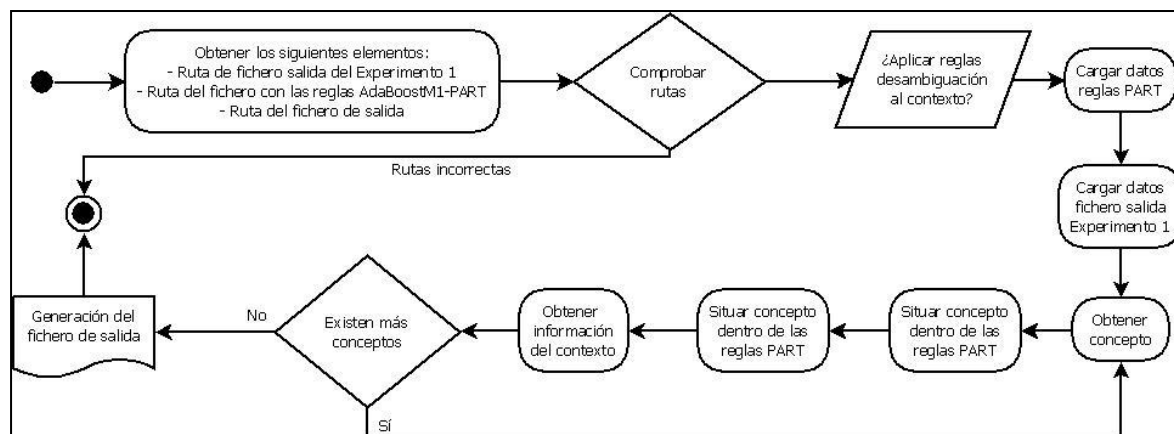


Figura 21: Diagrama de actividad de la creación del fichero de la Fase 2 de Experimentación

Debido a que se generan dos tipos de fichero la segunda fase de experimentación, se han realizado dos diagramas de clases para cada uno de ellos, diferenciándose únicamente en que la clase a la que se llama, siendo Experimento2 (las instancias que se obtienen constan de 5 atributos) o Experimento2\_3reglas (las instancias que se obtienen constan de 7 atributos):

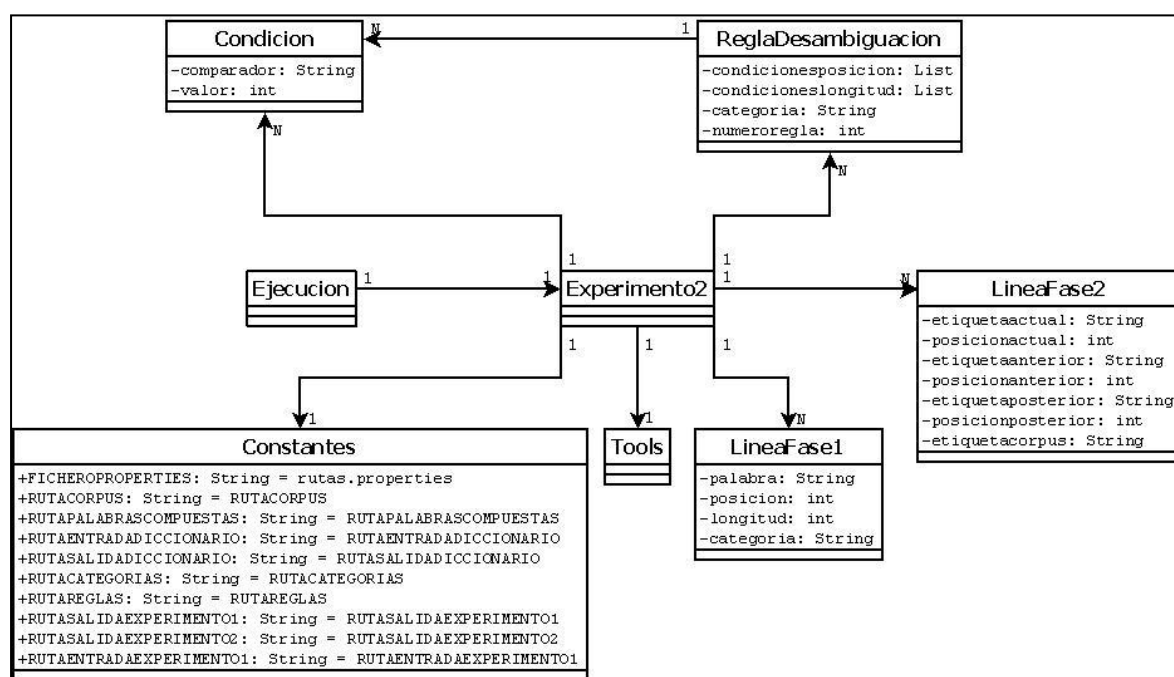


Figura 22: Diagrama de clases para la generación del fichero de la Fase 2 de experimentación (5 atributos)

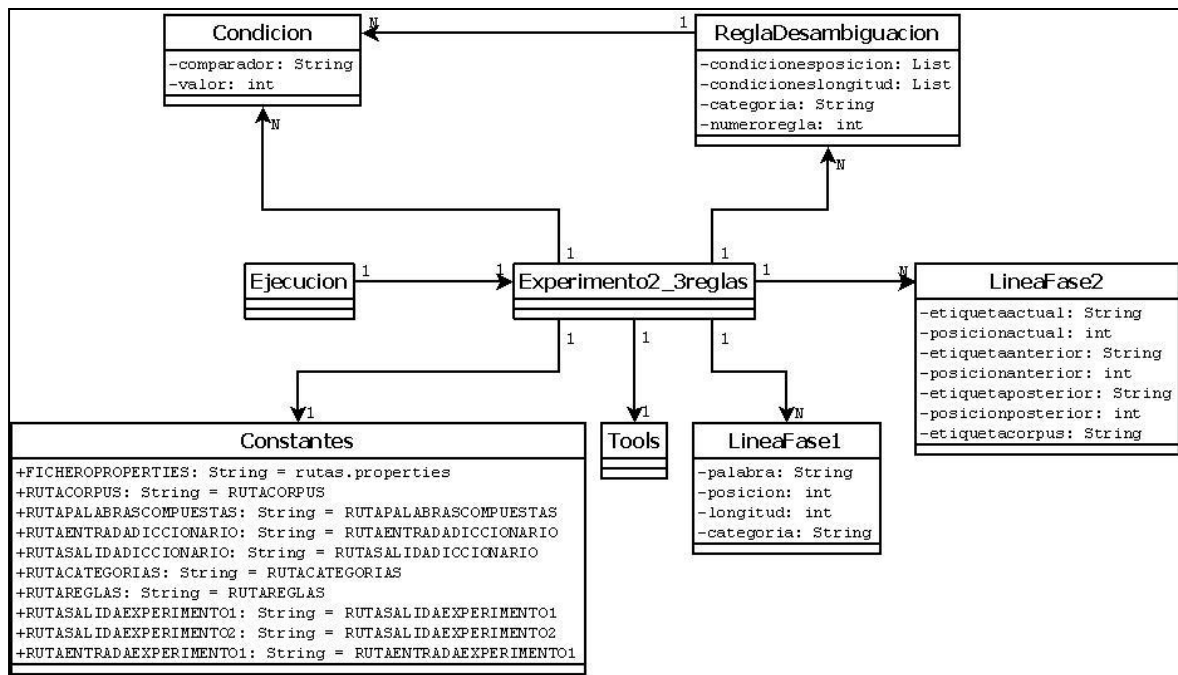


Figura 23: Diagrama de clases para la generación del fichero de la Fase 2 de experimentación (7 atributos)

## 4.2 Algoritmos de Clasificación de Weka

Se han efectuado experimentos utilizando algoritmos de clasificación diferentes para encontrar el de mayor efectividad y, de esa forma, conseguir unas reglas de desambiguación con mayor garantía de éxito. Los algoritmos con los que se ha clasificado son básicamente orientados a reglas, aunque también se ha utilizado el J48, que se basa en árboles de decisión. Además, se han efectuado experimentos con algoritmos o meta-algoritmos de *Bagging* y de *Boosting*, que pasan a describirse brevemente.

El Bagging (*Bootstrap aggregating*) es un algoritmo propuesto por Leo Breiman en 1994 que mejora las estadísticas dentro de un proceso de aprendizaje combinando clasificadores, normalmente aplicado a modelos de árbol de decisión, aunque pueden ser utilizados con cualquier otro tipo. Se basa en el *bootstrap*, un método de remuestreo propuesto por Bradley Efron en 1979, en el que se aproxima la distribución en el muestreo de una estadística. Es usado de manera frecuente para aproximar sesgo y varianza de una estadística, construir intervalos de confianza o contrastar hipótesis sobre parámetros de interés. Debido a que requiere de cálculos y remuestras para realizar aproximaciones, se hace necesario el uso de un ordenador de gran potencia para poner en práctica este método.

La idea original del Boosting fue propuesta por Robert Schapire y Yoav Freund en 1996 para reducir el sesgo en el aprendizaje supervisado. La mayoría de los algoritmos de Boosting consisten en el aprendizaje de forma iterativa de clasificadores débiles con respecto a una distribución y la adición de ellos a un clasificador fuerte final. Al principio se asigna un peso para todos y en cada iteración se construye un clasificador base teniendo en cuenta la distribución de pesos. Poco después, los pesos de cada ejemplo se reajustan acorde a la clasificación correcta o incorrecta que se haya dado. En la clasificación final,

el resultado se obtendrá mediante votos ponderados de los clasificadores base.

Los algoritmos utilizados en este proyecto son los siguientes:

- Reglas: ConjunctiveRule, DecisionTable, OneR, Part, Ridor y ZeroR
- Árboles: J48
- Meta-algoritmos:
  - Bagging: Bagging (con J48 y Part)
  - Boosting: AdaBoostM1 (con J48 y Part) y MultiBoostAB (con J48)

## 5. EXPERIMENTACIÓN 0: Experimentos previos

Los primeros experimentos, pruebas para familiarizarse con el entorno de trabajo, la terminología y los objetivos, se realizaron sobre el corpus completo y sin mapear las categorías gramaticales que en él aparecen. Posteriormente, se toma la determinación de realizar un mapeo para obtener las categorías gramaticales generales a partir de las etiquetas del corpus. Para estos experimentos se utilizó un diccionario de sólo 918 palabras diferentes, lo cual se comprobó que no era muy eficiente para desambiguar dada su escasa variabilidad. Además, no se utilizó el total del corpus, sino las muestras de la carpeta *news* (noticias), por motivos de carga de datos. Los resultados de estos primeros experimentos, especificando el tipo de clasificación utilizado en cada caso y su porcentaje de acierto, fueron:

### Fase 1:

ZeroR: 30,3143%

ConjunctiveRule: 39,7463%

OneR: 79,1315%

Ridor: 87,5091%

DecisionTable: 88,2629%

J48: 88,8461%

**Part: 88,8918%**

### Fase 2 (Contando con las reglas de Part y sin la posición de las palabras del contexto):

ZeroR: 30,3149%

ConjunctiveRule: 40,6874%

OneR: 89,6285%

DecisionTable: 90,0342%

Ridor: 90,4826%

J48: 91,5409%

**Part: 91,5816%**

### Fase 2 (Contando con las reglas de Part y la posición de las palabras del contexto):

ZeroR: 30,3149%

ConjunctiveRule: 40,6874%

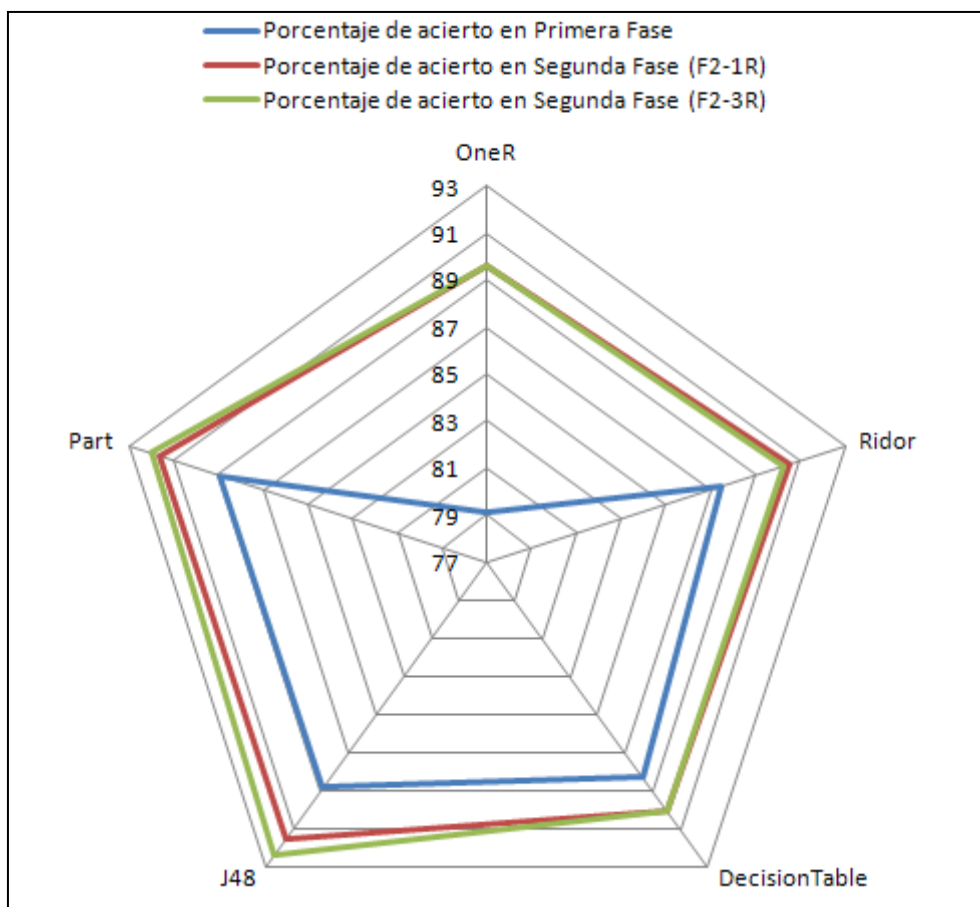
OneR: 89,6285%

DecisionTable: 90,0551%

Ridor: 90,2594%

Part: 92,0210%

**J48: 92,3861%**



**Figura 24: Gráfica comparativa de los algoritmos con más éxito en los experimentos previos**

Como se puede apreciar, los algoritmos Rules Part y Tree J48 ya se postulan desde el inicio como los de mayor efectividad en ambas fases. Se debe tener en cuenta que los resultados obtenidos también están condicionados por la inclusión de datos alfanuméricos, lo cual puede repercutir negativamente. Para los experimentos relevantes se han obviado dichos tokens, además de los símbolos de puntuación, de manera que solo se realiza la desambiguación con caracteres alfabéticos, utilizando para ello un diccionario compuesto por todas las palabras diferentes del corpus, superando los 40.000 términos.

## 6. EXPERIMENTACIÓN 1: Brown corpus, vocablos ingleses

En total se han realizado 36 experimentos diferentes a lo largo de ambas fases (12 para la primera fase y 24 para la segunda fase de experimentación), uno con cada tipo de algoritmo de clasificación. A continuación se detallan los resultados de cada uno de dichos ensayos, ordenados de menor a mayor porcentaje de éxito.

### 6.1 Primera Fase de Experimentación

#### 6.1.1 Experimento 1

En el Experimento 1 se efectúa la clasificación con el algoritmo **ZeroR**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.ZeroR
Correctly Classified Instances    258751    26.1246 %
Incorrectly Classified Instances  731697    73.8754 %
Kappa statistic                  0
Mean absolute error              0.1562
Root mean squared error         0.2795
Relative absolute error          100 %
Root relative squared error      100 %
Total Number of Instances       990448

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0        0        0          0        0         0.5     adjective
      0        0        0          0        0         0.5     adverb
      0        0        0          0        0         0.5     aux
      0        0        0          0        0         0.5     determiner
      0        0        0          0        0         0.5     invariant
      1        1        0.261      1        0.414     0.5     noun
      0        0        0          0        0         0.5     preposition
      0        0        0          0        0         0.5     pronoun
      0        0        0          0        0         0.5     qualifier
      0        0        0          0        0         0.5     quantifier
      0        0        0          0        0         0.5     verb
Weighted Avg.  0.261    0.261    0.068    0.261    0.108    0.5

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a   b   c   d   e   f   g   h   i   j   k  <-- classified as
0   0   0   0   0   68476  0   0   0   0   0 | a = adjective
0   0   0   0   0   55115  0   0   0   0   0 | b = adverb
0   0   0   0   0   69523  0   0   0   0   0 | c = aux
0   0   0   0   0   138031  0   0   0   0   0 | d = determiner
0   0   0   0   0   75748  0   0   0   0   0 | e = invariant
0   0   0   0   0   258751  0   0   0   0   0 | f = noun
0   0   0   0   0   122437  0   0   0   0   0 | g = preposition
0   0   0   0   0   73521  0   0   0   0   0 | h = pronoun
0   0   0   0   0   9541  0   0   0   0   0 | i = qualifier
0   0   0   0   0   3752  0   0   0   0   0 | j = quantifier
0   0   0   0   0   115553  0   0   0   0   0 | k = verb

```

Como se puede observar en los resultados, el algoritmo ZeroR queda descartado totalmente para ser elegido como generador de reglas de desambiguación, ya que clasifica todas las palabras como sustantivos (*noun*), lo que explica el bajo porcentaje de acierto.



## 6.1.2 Experimento 2

Para el Experimento 2 se ejecuta la clasificación con el algoritmo **ConjunctiveRule**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Correctly Classified Instances      347426      35.0777 %
Incorrectly Classified Instances    643022      64.9223 %
Kappa statistic                    0.1783
Mean absolute error                0.1433
Root mean squared error            0.2677
Relative absolute error             91.7624 %
Root relative squared error         95.795 %
Total Number of Instances          990448

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0         0         0         0         0         0.689     adjective
      0         0         0         0         0         0.566     adverb
      0         0         0         0         0         0.639     aux
      0.761     0.367     0.252     0.761     0.378     0.697     determiner
      0         0         0         0         0         0.714     invariant
      0.937     0.452     0.423     0.937     0.583     0.742     noun
      0         0         0         0         0         0.705     preposition
      0         0         0         0         0         0.684     pronoun
      0         0         0         0         0         0.537     qualifier
      0         0         0         0         0         0.654     quantifier
      0         0         0         0         0         0.701     verb
Weighted Avg.   0.351     0.169     0.146     0.351     0.205     0.697

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a    b    c    d    e    f    g    h    i    j    k  <-- classified as
0    0    0  4760    0  63716    0    0    0    0    0 | a = adjective
0    0    0 16354    0  38761    0    0    0    0    0 | b = adverb
0    0    0 47456    0 22067    0    0    0    0    0 | c = aux
0    0    0105058    0 32973    0    0    0    0    0 | d = determiner
0    0    0 62092    0 13656    0    0    0    0    0 | e = invariant
0    0    0 16383    0 242368    0    0    0    0    0 | f = noun
0    0    0 95677    0 26760    0    0    0    0    0 | g = preposition
0    0    0 56094    0 17427    0    0    0    0    0 | h = pronoun
0    0    0 3290    0 6251    0    0    0    0    0 | i = qualifier
0    0    0 2758    0 994    0    0    0    0    0 | j = quantifier
0    0    0 7584    0107969    0    0    0    0    0 | k = verb

```

A la luz de los resultados, no se ha mejorado mucho con respecto al algoritmo ZeroR, pues en este caso clasifica las palabras como sustantivos (*noun*) o como determinantes (*determiner*) y queda también descartado para ser el elegido como generador de reglas de desambiguación. El porcentaje de acierto es aún alarmantemente bajo.

### 6.1.3 Experimento 3

El algoritmo de clasificación utilizado en el Experimento 3 es el llamado **Ridor**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.Ridor -F 3 -S 1 -N 2.0
Correctly Classified Instances      924280      93.3194 %
Incorrectly Classified Instances    66168      6.6806 %
Kappa statistic                     0.9223
Mean absolute error                  0.0121
Root mean squared error              0.1102
Relative absolute error              7.7757 %
Root relative squared error          39.4352 %
Total Number of Instances          990448

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.911     0.008     0.893     0.911     0.902     0.951     adjective
      0.855     0.006     0.893     0.855     0.874     0.925     adverb
      0.999      0       0.994     0.999     0.996     0.999     aux
      0.964     0.003     0.979     0.964     0.971     0.98      determiner
      0.975     0.021     0.793     0.975     0.874     0.977     invariant
      0.951     0.018     0.949     0.951     0.95      0.967     noun
      0.885     0.005     0.962     0.885     0.922     0.94      preposition
      0.967     0.001     0.989     0.967     0.978     0.983     pronoun
      0.486     0.004     0.55      0.486     0.516     0.741     qualifier
      0.993      0       0.93      0.993     0.96      0.996     quantifier
      0.905     0.009     0.928     0.905     0.916     0.948     verb
Weighted Avg.   0.933     0.009     0.936     0.933     0.933     0.962

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
62363  1219      3    343     61   3051     77      6    282      0   1071 | a = adjective
1382   47133    74    874    471    203   3006    571   1215      8    178 | b = adverb
2       2   69448      1      2     20      0      0      0      1     47 | c = aux
193    397      0  133028   2557    100      0     16   1739      0      1 | d = determiner
12     363      3     13   73817    123    895     10    480      0     32 | e = invariant
4304    551    324    278    269  246144    214    156      47    19   6445 | f = noun
38    1164      0     45  12671    108  108341      0     32      0     38 | g = preposition
2       2      2    570   1817     24      2   71098      0      0      4 | h = pronoun
528    1810      0    704   1158     80     51      0   4638    254    318 | i = qualifier
0       0      0     25      0      1      0      0      0   3726      0 | j = quantifier
1023    136     22     33    321   9388     74      7      5      0  104544 | k = verb

```

Observando los resultados obtenidos con el algoritmo Ridor, se puede comprobar una importante mejoría en los aciertos ya que ahora las palabras se clasifican en todas las clases, aunque todavía con errores. Se puede conseguir mayor porcentaje de instancias correctas, por lo que continúa la búsqueda de un algoritmo de mayor eficacia.

## 6.1.4 Experimento 4

En el Experimento 4 se efectúa la clasificación con el algoritmo **OneR**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.OneR -B 6
Correctly Classified Instances      925068      93.3989 %
Incorrectly Classified Instances    65380       6.6011 %
Kappa statistic                    0.9232
Mean absolute error                 0.012
Root mean squared error             0.1096
Relative absolute error             7.6831 %
Root relative squared error         39.1997 %
Total Number of Instances          990448
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.906    0.008    0.895    0.906    0.9        0.949    adjective
      0.854    0.005    0.902    0.854    0.878      0.924    adverb
      0.999     0      0.996    0.999    0.997      0.999    aux
      0.964    0.003    0.979    0.964    0.971      0.98     determiner
      0.974    0.021    0.795    0.974    0.875      0.977    invariant
      0.956    0.02    0.945    0.956    0.95       0.968    noun
      0.885    0.005    0.963    0.885    0.922      0.94     preposition
      0.967    0.001    0.99     0.967    0.978      0.983    pronoun
      0.52     0.004    0.579    0.52     0.548      0.758    qualifier
      0.993     0      0.93     0.993    0.961      0.996    quantifier
      0.902    0.009    0.933    0.902    0.917      0.947    verb
Weighted Avg.    0.934    0.01    0.936    0.934    0.934      0.962
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
62033  770      1      344      50      3702    125      6      48      0    1397 | a = adjective
1508  47074      0      876      470      268    3004    569    1311      8      27 | b = adverb
2      0  69437      1      2      56      0      1      0      0      24 | c = aux
193    398      0  133004    2555    127      0    15    1739      0      0 | d = determiner
19    365      1      12    73815    151    853      6    480      0     46 | e = invariant
4201    392    298    232    118  247336     91     66     27    18    5972 | f = noun
39   1143      0     45   12674    128  108356      0      0      0     52 | g = preposition
4      3      0     570    1817     46      1   71074      0      0      6 | h = pronoun
541   1879      0     704    1158     31      1      0   4965    254      8 | i = qualifier
1      0      0     25      0      0      0      0      0   3726      0 | j = quantifier
801    148      8     45     223   10003     58     19      0      0  104248 | k = verb
```

Con el algoritmo OneR se ha conseguido una leve mejoría con respecto a Ridor, pero aún puede incrementarse el porcentaje de aciertos, por lo que se procede a la ejecución de un nuevo experimento con otro algoritmo diferente.

## 6.1.5 Experimento 5

Para el Experimento 5 se ejecuta la clasificación con el algoritmo **DecisionTable**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Correctly Classified Instances      926321                93.5255 %
Incorrectly Classified Instances    64127                 6.4745 %
Kappa statistic                    0.9247
Mean absolute error                 0.0217
Root mean squared error             0.095
Relative absolute error             13.9019 %
Root relative squared error         34.0046 %
Total Number of Instances          990448

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.913    0.008    0.896    0.913    0.904    0.995    adjective
      0.857    0.005    0.903    0.857    0.879    0.995    adverb
      0.999     0      0.996    0.999    0.997     1      aux
      0.964    0.003    0.979    0.964    0.971    0.999    determiner
      0.975    0.021    0.795    0.975    0.876    0.995    invariant
      0.958    0.019    0.947    0.958    0.952    0.995    noun
      0.885    0.005    0.964    0.885    0.923    0.998    preposition
      0.967    0.001    0.991    0.967    0.979     1      pronoun
      0.505    0.004    0.583    0.505    0.541    0.989    qualifier
      0.993     0      0.93    0.993    0.961     1      quantifier
      0.904    0.008    0.934    0.904    0.919    0.994    verb
Weighted Avg.    0.935    0.01    0.938    0.935    0.935    0.997

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
62509  765      2    325     50   3221    106      0     47      0   1451 | a = adjective
1511  47209      1    893    464    190   3007    570   1257      8      5 | b = adverb
4      2  69435      0      2     49      0      0      0      0     31 | c = aux
194    399      0 133119   2556    107      0     15   1641      0      0 | d = determiner
23    361      1     12  73830    136    854      4    480      0     47 | e = invariant
4150   341    284    188    109 247757     60     52     21    18   5771 | f = noun
41   1145      0     45 12676   119 108370      0      0      0     41 | g = preposition
1      3      0    569   1820     38      0  71088      0      0      2 | h = pronoun
542   1919      0    826   1158     17      1      0   4816    254      8 | i = qualifier
0      0      0     25      0      1      0      0      0   3726      0 | j = quantifier
781    115      4     28    221   9902     40      0      0      0 104462 | k = verb

```

Se ha incrementado en más de un 0,12% el porcentaje de acierto con respecto al anterior algoritmo. Aún así, se van a seguir probando algoritmos de clasificación en pos de un mejor resultado.

## 6.1.6 Experimento 6

El algoritmo de clasificación utilizado en el Experimento 6 es el **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Correctly Classified Instances      927539                93.6484 %
Incorrectly Classified Instances    62909                 6.3516 %
Kappa statistic                    0.9261
Mean absolute error                 0.0164
Root mean squared error             0.092
Relative absolute error             10.5242 %
Root relative squared error         32.9309 %
Total Number of Instances          990448

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.923    0.008    0.901    0.923    0.912    0.994    adjective
      0.857    0.005    0.904    0.857    0.88    0.995    adverb
      0.999    0      0.996    0.999    0.997    1      aux
      0.964    0.003    0.979    0.964    0.972    0.999    determiner
      0.975    0.021    0.795    0.975    0.876    0.995    invariant
      0.957    0.018    0.95    0.957    0.954    0.994    noun
      0.885    0.005    0.964    0.885    0.923    0.998    preposition
      0.967    0.001    0.991    0.967    0.979    1      pronoun
      0.52     0.004    0.581    0.52     0.549    0.99    qualifier
      0.993    0      0.929    0.993    0.96    1      quantifier
      0.908    0.008    0.935    0.908    0.921    0.993    verb
Weighted Avg.    0.936    0.009    0.939    0.936    0.937    0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63232    716      0    332     50   2842    106      1     47      2   1148 | a = adjective
1497   47244      1    875     46     158   3000   571   1281      9     13 | b = adverb
2         0   69443      1      1     40      0      2      0      1     33 | c = aux
194     397      0 133027   2557    101      0    15   1739      0      1 | d = determiner
14     363      4     13   73839    126    855      6    481      0     47 | e = invariant
3902    355    289    194    134 247735      82    71     31    20   5938 | f = noun
38   1142      0     45  12678    111 108366      5      0      0     52 | g = preposition
0         6      2    570   1818     25      3  71095      0      0      2 | h = pronoun
540   1900      0    704   1159     16      1      0   4962    254      5 | i = qualifier
0         0      0     25      0      1      0      0      0   3726      0 | j = quantifier
763    115      8     27    220   9513     29      8      0      0 104870 | k = verb

```

Se ha vuelto incrementado en más de un 0,12% el porcentaje de acierto con respecto al anterior algoritmo (DecisionTable). Queda por probar el algoritmo J48 y algunos meta-algoritmos, que consiguen superar los resultados obtenidos hasta el momento.

## 6.1.7 Experimento 7

En el Experimento 7 se efectúa la clasificación con el algoritmo de árboles **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Correctly Classified Instances      928376                93.7329 %
Incorrectly Classified Instances    62072                  6.2671 %
Kappa statistic                    0.9271
Mean absolute error                 0.0165
Root mean squared error             0.0922
Relative absolute error             10.5652 %
Root relative squared error         32.9806 %
Total Number of Instances          990448
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.924     0.007     0.903     0.924     0.913       0.993     adjective
      0.857     0.005     0.904     0.857     0.88        0.995     adverb
      0.999      0       0.995     0.999     0.997       1         aux
      0.964     0.003     0.98      0.964     0.972       0.999     determiner
      0.975     0.021     0.795     0.975     0.876       0.995     invariant
      0.96      0.017     0.951     0.96      0.955       0.994     noun
      0.885     0.005     0.964     0.885     0.923       0.998     preposition
      0.967     0.001     0.991     0.967     0.979       1         pronoun
      0.52      0.004     0.582     0.52      0.549       0.99      qualifier
      0.993      0       0.93      0.993     0.96        1         quantifier
      0.91      0.008     0.94      0.91      0.925       0.992     verb
Weighted Avg.   0.937     0.009     0.94      0.937     0.938       0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63286   742      3    329     48   2902    110      1     48      0   1007 | a = adjective
1490  47217      0    876    465    203   3002    569   1272      8     13 | b = adverb
      2      0  69440      1      1     28      0      1      2      1     47 | c = aux
194    397      0 133028   2556    101      0    15   1739      0      1 | d = determiner
14    361      1     13  73838    138    855      5    480      0     43 | e = invariant
3786   328    300    189    133 248287     86    59     21    19   5543 | f = noun
38   1143      0     45 12676   110 108366      2      0      0     57 | g = preposition
      0      3      2    573   1818     27      3  71091      0      0      4 | h = pronoun
542   1899      0    704   1158     18      1      0  4959   254      6 | i = qualifier
      0      0      0     25      0      1      0      0    3726      0 | j = quantifier
736    114      9     29    219   9275     29      3      1      0 105138 | k = verb
```

Esta vez la mejora con el anterior algoritmo (Part) queda por debajo del 0,1%. Hasta aquí llegan los experimentos con algoritmos de árboles y reglas, los siguientes serán clasificados con meta-algoritmos de bagging y boosting, buscando un mayor acierto en la etiquetación.

## 6.1.8 Experimento 8

Para el Experimento 8 se ejecuta la clasificación con el meta-algoritmo **Bagging** basado en el **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      928524                93.7479 %
Incorrectly Classified Instances    61924                 6.2521 %
Kappa statistic                    0.9272
Mean absolute error                0.0165
Root mean squared error            0.091
Relative absolute error            10.5383 %
Root relative squared error        32.5687 %
Total Number of Instances          990448

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.923    0.007    0.905    0.923    0.914    0.996    adjective
      0.857    0.005    0.904    0.857    0.88    0.995    adverb
      0.999     0      0.995    0.999    0.997    1        aux
      0.964    0.003    0.979    0.964    0.972    0.999    determiner
      0.975    0.021    0.795    0.975    0.876    0.995    invariant
      0.961    0.018    0.95    0.961    0.955    0.995    noun
      0.885    0.005    0.964    0.885    0.923    0.998    preposition
      0.967    0.001    0.991    0.967    0.979    1        pronoun
      0.508    0.004    0.585    0.508    0.544    0.99    qualifier
      0.993     0      0.93    0.993    0.96    1        quantifier
      0.909    0.007    0.942    0.909    0.925    0.994    verb
Weighted Avg.    0.937    0.009    0.94    0.937    0.938    0.997

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63191  753      4    329     48   3016    110      1     51      0   973 | a = adjective
1464  47261     0    887    465    205   2998    568   1246     8   13 | b = adverb
1      0  69441     1      1     39      0      1      1     1   37 | c = aux
194   397      0  133121  2556    106      0    15   1640     0     2 | d = determiner
12    360      2     13  73836    139    854      7    480     0    45 | e = invariant
3662   330    298    195    129  248572    81    58     21    19  5386 | f = noun
38   1144      0     45  12676    111  108364      2      0     0    57 | g = preposition
0      3      2     573   1818     28      3  71091      0     0     3 | h = pronoun
541   1917     0    799   1158     20      1      0   4845    254     6 | i = qualifier
0      0      0     25      0      1      0      0      0  3726     0 | j = quantifier
696   112     12     28    220   9379      27      3      0      0  105076 | k = verb

```

El porcentaje de acierto utilizando el Bagging sobre J48 no se ha incrementado ni 0,02% con respecto al J48, algo prácticamente despreciable. Se probó con el algoritmo Part como base para el Bagging para intentar conseguir un mayor número de aciertos.

## 6.1.9 Experimento 9

El meta-algoritmo de clasificación utilizado en el Experimento 9 es el **MultiBoostAB**, apoyado en el algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.MultiBoostAB -C 3 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      928664                93.762 %
Incorrectly Classified Instances    61784                  6.238 %
Kappa statistic                    0.9274
Mean absolute error                 0.0115
Root mean squared error             0.1041
Relative absolute error              7.3313 %
Root relative squared error         37.2545 %
Total Number of Instances          990448
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.926     0.007     0.905     0.926     0.915       0.993     adjective
      0.856     0.005     0.905     0.856     0.88        0.993     adverb
      0.999     0       0.996     0.999     0.997       1         aux
      0.964     0.003     0.98      0.964     0.972       0.999     determiner
      0.974     0.021     0.795     0.974     0.875       0.994     invariant
      0.96      0.017     0.952     0.96      0.956       0.995     noun
      0.885     0.005     0.963     0.885     0.923       0.997     preposition
      0.967     0.001     0.991     0.967     0.979       1         pronoun
      0.523     0.004     0.578     0.523     0.549       0.98      qualifier
      0.993     0       0.93      0.993     0.96        0.999     quantifier
      0.911     0.008     0.94      0.911     0.925       0.991     verb
Weighted Avg.   0.938     0.009     0.94      0.938     0.938       0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63392   776      3    329     47   2841    108      1     51      0   928 | a = adjective
1458   47191      0    876    450    178   3028    569   1348      8     9 | b = adverb
1         0   69444      1      1     45      0      1      2      1    27 | c = aux
194     397      0 133030   2556     99      0    15   1739      0     1 | d = determiner
11     361      1     13   73795    131    897      9    480      0    50 | e = invariant
3708    327    293    189    139 248333     82    57     21    19   5583 | f = noun
36    1145      0     45  12648    109 108401      2      0      0    51 | g = preposition
0         2      0     573   1820     25      1  71096      0      0     4 | h = pronoun
540    1851      0    704   1158     18     16      0   4992    254     8 | i = qualifier
0         0      0     25      0      1      0      0      0   3726     0 | j = quantifier
719     106      6     27    220   9170     35      5      1      0 105264 | k = verb
```

Esta vez, con este meta-algoritmo de boosting, se ha conseguido una mejora de 0,03% con respecto al J48 básico. Queda por realizar experimentos con otro algoritmo de boosting (AdaBoostM1) y el Bagging tomando como referencia el algoritmo Part.



## 6.1.10 Experimento 10

En el Experimento 10 se efectúa la clasificación con el meta-algoritmo **Bagging**, tomando como base el algoritmo **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      928762                93.7719 %
Incorrectly Classified Instances    61686                 6.2281 %
Kappa statistic                    0.9275
Mean absolute error                 0.0164
Root mean squared error             0.0903
Relative absolute error             10.4929 %
Root relative squared error         32.317 %
Total Number of Instances          990448
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.926     0.007     0.904     0.926     0.915     0.996     adjective
      0.858     0.005     0.905     0.858     0.881     0.995     adverb
      0.999     0       0.996     0.999     0.997     1         aux
      0.964     0.003     0.979     0.964     0.972     0.999     determiner
      0.975     0.021     0.795     0.975     0.876     0.995     invariant
      0.96      0.017     0.952     0.96      0.956     0.996     noun
      0.885     0.005     0.964     0.885     0.923     0.998     preposition
      0.967     0.001     0.991     0.967     0.979     1         pronoun
      0.509     0.004     0.585     0.509     0.545     0.991     qualifier
      0.993     0       0.929     0.993     0.96      1         quantifier
      0.911     0.008     0.94      0.911     0.925     0.995     verb
Weighted Avg.   0.938     0.009     0.94      0.938     0.938     0.997
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63382   721      0   331     48   2783   108      1     46      0  1056 |   a = adjective
1492  47270      0   884    463   165  2999   570  1254      8    10 |   b = adverb
  1      0  69444      2      1     39      0      2      0      1    33 |   c = aux
 194   398      0 133124  2558   101      0     15  1640      0     1 |   d = determiner
 12   361      4     13  73841   129   855      6   480      0    47 |   e = invariant
3740   319   293   194   128 248443   76    55    19    20  5464 |   f = noun
 38  1144      0    45  12677   112 108366      2      0      0    53 |   g = preposition
  0      6      2   570  1817    26      3  71095      0      0      2 |   h = pronoun
541  1909      0   799  1159    16      1      0  4856   254     6 |   i = qualifier
  0      0      0    25      0      1      0      0      0  3726     0 |   j = quantifier
688   112      6     26   219   9256    26      5      0      0 105215 |   k = verb
```

La aplicación del meta-algoritmo de bagging al algoritmo Part supone una mejora de más del 0,12% sobre el Part normal. Como se está viendo en estos últimos experimentos, las mejoras son relativamente pobres, el algoritmo elegido esta cercano.

## 6.1.11 Experimento 11

Para el Experimento 11 se ejecuta la clasificación con el meta-algoritmo **AdaBoostM1** basado en el **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      929124                93.8085 %
Incorrectly Classified Instances    61324                  6.1915 %
Kappa statistic                     0.928
Mean absolute error                 0.0233
Root mean squared error             0.0919
Relative absolute error             14.9029 %
Root relative squared error         32.8712 %
Total Number of Instances          990448
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.93      0.007     0.905     0.93     0.917     0.994     adjective
      0.857     0.005     0.905     0.857     0.88     0.995     adverb
      0.999      0       0.996     0.999     0.997     1         aux
      0.964     0.003     0.98     0.964     0.972     0.999     determiner
      0.975     0.021     0.795     0.975     0.876     0.995     invariant
      0.96      0.017     0.953     0.96     0.956     0.995     noun
      0.885     0.005     0.964     0.885     0.923     0.998     preposition
      0.967     0.001     0.991     0.967     0.979     1         pronoun
      0.527     0.004     0.578     0.527     0.552     0.986     qualifier
      0.993      0       0.929     0.993     0.96     1         quantifier
      0.912     0.008     0.941     0.912     0.926     0.993     verb
Weighted Avg.   0.938     0.009     0.941     0.938     0.938     0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63679   795      0    331     48   2664    104      1     50      0   804 | a = adjective
1434  47234      1    874    464    128   3019    570   1376      8     7 | b = adverb
      4      0  69449      0      1     35      0      1      0      1    32 | c = aux
194    397      0 133031   2557     97      0    15   1739      0     1 | d = determiner
11     362      1     13  73866    109    855      7    481      0    43 | e = invariant
3742   321    282    187    119 248274     65    46     24    20  5671 | f = noun
36   1138      0     45  12676    106 108393      2      0      0    41 | g = preposition
      1      5      2    570   1819     17      0  71104      0      0     3 | h = pronoun
538   1833      0    704   1159     16      1      0   5031    254     5 | i = qualifier
      0      0      0     25      0      1      0      0      0  3726     0 | j = quantifier
756    114      8     26    221   9041     44      5      1      0 105337 | k = verb
```

El porcentaje de acierto utilizando este meta-algoritmo de boosting sobre Part se ha incrementado en algo más de 0,16% con respecto al Part normal. Este es un claro candidato a ser el algoritmo elegido para obtener las reglas de desambiguación.

## 6.1.12 Experimento 12

El meta-algoritmo de clasificación utilizado en el Experimento 12 es el **AdaBoostM1**, apoyado en el algoritmo **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      929548                93.8513 %
Incorrectly Classified Instances    60900                  6.1487 %
Kappa statistic                    0.9285
Mean absolute error                 0.0231
Root mean squared error             0.0916
Relative absolute error             14.8133 %
Root relative squared error         32.7733 %
Total Number of Instances          990448

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.931    0.007    0.906    0.931    0.918    0.993    adjective
      0.856    0.005    0.906    0.856    0.88    0.995    adverb
      0.999    0      0.996    0.999    0.997    1        aux
      0.964    0.003    0.98    0.964    0.972    0.999    determiner
      0.975    0.021    0.795    0.975    0.876    0.995    invariant
      0.959    0.016    0.955    0.959    0.957    0.995    noun
      0.885    0.005    0.964    0.885    0.923    0.998    preposition
      0.967    0.001    0.991    0.967    0.979    1        pronoun
      0.53     0.004    0.575    0.53     0.552    0.985    qualifier
      0.993    0      0.929    0.993    0.96     1        quantifier
      0.915    0.008    0.941    0.915    0.928    0.993    verb
Weighted Avg.    0.939    0.009    0.941    0.939    0.939    0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63769    791      2    329      48   2656    105      1     51      0   724 | a = adjective
1421  47176      0    876    463    141   3011   569   1444      8     6 | b = adverb
      2      0  69449      1      2     17      0      1      2      1    48 | c = aux
194    397      0 133030   2558     96      0     15   1739      0     2 | d = determiner
12    360      1     13  73865    112    855      7    480      0    43 | e = invariant
3680    319    283    185    116 248197     71    49     21    20   5810 | f = noun
38   1136      0     45 12676    107 108392      2      0      0    41 | g = preposition
      0      3      2   573   1818     16      0  71106      0      0     3 | h = pronoun
537   1803      0    704   1158     18      1      0   5061    254     5 | i = qualifier
      0      0      0     25      0      1      0      0      0  3726     0 | j = quantifier
759    112      7     28    218   8599     50      3      0      0 105777 | k = verb

```

Con este meta-algoritmo de boosting, se ha conseguido una mejora de casi 0,22% con respecto al J48 básico. Tanto este como el mismo meta-algoritmo basado en el Part son los dos máximos candidatos a ser elegidos para crear las reglas de desambiguación.

### 6.1.13 Conclusiones y resultados de la Primera Fase de Experimentación

A la vista de los resultados obtenidos con todos los algoritmos, se decide escoger el meta-algoritmo AdaBoostM1 ejecutado sobre la clasificación Part, dado que es uno de los que mejores resultados obtiene y es un modelo basado en reglas, mientras que el AdaBoostM1 ejecutado sobre J48 el resultado viene expresado en forma de árbol de decisión. Con las reglas de desambiguación generadas, se espera mejorar los resultados en una segunda fase, donde también se tendrá en cuenta el contexto.

A continuación se muestra una gráfica comparando los porcentajes de acierto de los algoritmos utilizados en la Primera Fase de Experimentación:

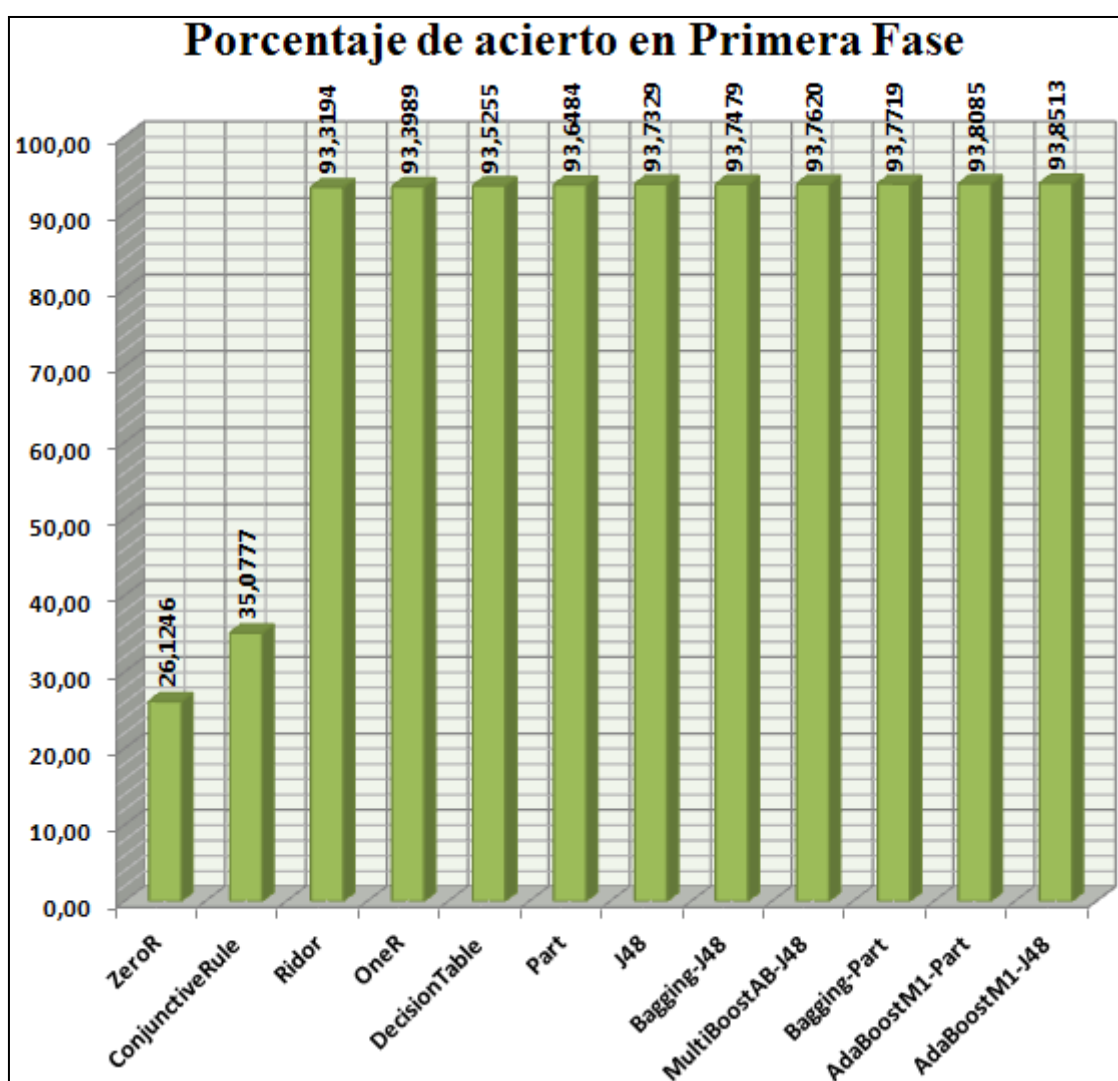


Figura 25: Gráfica de comparación de los algoritmos en la 1ª fase de experimentación

## 6.2 Segunda Fase de Experimentación

### 6.2.1 Experimentos con número de regla sólo para la palabra (F2-1R)

#### 6.2.1.1 Experimento 1

En el Experimento 1 se efectúa la clasificación con el algoritmo **ZeroR**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.ZeroR
Correctly Classified Instances      258749      26.1245 %
Incorrectly Classified Instances    731697      73.8755 %
Kappa statistic                     0
Mean absolute error                 0.1562
Root mean squared error             0.2795
Relative absolute error              100      %
Root relative squared error          100      %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0         0         0         0         0         0.5  adjective
          0         0         0         0         0         0.5  adverb
          0         0         0         0         0         0.5  aux
          0         0         0         0         0         0.5  determiner
          0         0         0         0         0         0.5  invariant
          1         1       0.261         1       0.414         0.5  noun
          0         0         0         0         0         0.5  preposition
          0         0         0         0         0         0.5  pronoun
          0         0         0         0         0         0.5  qualifier
          0         0         0         0         0         0.5  quantifier
          0         0         0         0         0         0.5  verb
Weighted Avg.    0.261    0.261    0.068    0.261    0.108    0.5
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a   b   c   d   e   f   g   h   i   j   k  <-- classified as
0   0   0   0   0  68476  0   0   0   0   0 | a = adjective
0   0   0   0   0  55115  0   0   0   0   0 | b = adverb
0   0   0   0   0  69523  0   0   0   0   0 | c = aux
0   0   0   0   0 138031  0   0   0   0   0 | d = determiner
0   0   0   0   0  75748  0   0   0   0   0 | e = invariant
0   0   0   0   0 258749  0   0   0   0   0 | f = noun
0   0   0   0   0 122437  0   0   0   0   0 | g = preposition
0   0   0   0   0  73521  0   0   0   0   0 | h = pronoun
0   0   0   0   0  9541  0   0   0   0   0 | i = qualifier
0   0   0   0   0  3752  0   0   0   0   0 | j = quantifier
0   0   0   0   0 115553  0   0   0   0   0 | k = verb
```

El resultado no mejora el porcentaje obtenido por el mismo algoritmo en la primera fase, sino que lo iguala prácticamente.

### 6.2.1.2 Experimento 2

Para el Experimento 2 se ejecuta la clasificación con el algoritmo **ConjunctiveRule**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Correctly Classified Instances      387520          39.1258 %
Incorrectly Classified Instances    602926          60.8742 %
Kappa statistic                    0.1948
Mean absolute error                 0.1346
Root mean squared error             0.2594
Relative absolute error             86.1728 %
Root relative squared error         92.8284 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0         0         0         0         0         0.574    adjective
      0         0         0         0         0         0.57     adverb
      0         0         0         0         0         0.576    aux
      0.934     0.014     0.917     0.934     0.926     0.96     determiner
      0         0         0         0         0         0.529    invariant
      0.999     0.808     0.304     0.999     0.467     0.596    noun
      0         0         0         0         0         0.581    preposition
      0         0         0         0         0         0.563    pronoun
      0         0         0         0         0         0.543    qualifier
      0         0         0         0         0         0.571    quantifier
      0         0         0         0         0         0.58     verb
Weighted Avg.  0.391     0.213     0.207     0.391     0.251     0.63

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a   b   c   d   e   f   g   h   i   j   k  <-- classified as
0   0   0   233   0  68243   0   0   0   0   0 | a = adjective
0   0   0   552   0  54563   0   0   0   0   0 | b = adverb
0   0   0    0   0  69523   0   0   0   0   0 | c = aux
0   0   0 128936   0   9095   0   0   0   0   0 | d = determiner
0   0   0   6669   0  69079   0   0   0   0   0 | e = invariant
0   0   0   165   0  258584   0   0   0   0   0 | f = noun
0   0   0    3   0  122434   0   0   0   0   0 | g = preposition
0   0   0  1832   0   71689   0   0   0   0   0 | h = pronoun
0   0   0  2186   0   7355   0   0   0   0   0 | i = qualifier
0   0   0    0   0   3752   0   0   0   0   0 | j = quantifier
0   0   0    8   0 115545   0   0   0   0   0 | k = verb

```

En esta segunda fase sí se han incrementado el porcentaje de aciertos en más de un 4% con este algoritmo con respecto a los obtenidos con el mismo en la primera fase.

### 6.2.1.3 Experimento 3

El algoritmo de clasificación utilizado en el Experimento 3 es el llamado **OneR**.

Los resultados de la clasificación son:

```
Scheme: weka.classifiers.rules.OneR -B 6
Correctly Classified Instances      902998      91.1708 %
Incorrectly Classified Instances    87448      8.8292 %
Kappa statistic                    0.8972
Mean absolute error                 0.0161
Root mean squared error             0.1267
Relative absolute error             10.2764 %
Root relative squared error         45.3352 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.863    0.012    0.843    0.863    0.853    0.925    adjective
      0.844    0.006    0.894    0.844    0.868    0.919    adverb
      0.999     0      0.994    0.999    0.996    0.999    aux
      0.946    0.004    0.977    0.946    0.961    0.971    determiner
      0.968    0.023    0.776    0.968    0.862    0.973    invariant
      0.931    0.035    0.903    0.931    0.917    0.948    noun
      0.884    0.005    0.959    0.884    0.92    0.94    preposition
      0.966    0.001    0.99    0.966    0.978    0.982    pronoun
      0.499    0.004    0.531    0.499    0.514    0.747    qualifier
      0.993     0      0.929    0.993    0.96    0.996    quantifier
      0.826    0.012    0.899    0.826    0.861    0.907    verb
Weighted Avg.    0.912    0.015    0.915    0.912    0.912    0.948
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
59073  842      4      360      58    6110    111      0      7      4    1907 | a = adjective
1658  46538      0      924      530      287    3050    566    1528      8      26 | b = adverb
6      2    69437      0      1      41      3      0      0      0      33 | c = aux
199    388      0    130546    4514    170      48    15    2151      0      0 | d = determiner
35    365      1      12    73333    126    1301      3    510      0      62 | e = invariant
7612    702    405    285    136  240883      97    98    12    18    8501 | f = noun
43    1156      1      46    12662    118  108289      2      0      0    120 | g = preposition
18      4      0    561    1827    108      1   70998      0      0      4 | h = pronoun
548    1905      0    819    1158      88      1      0   4759    254      9 | i = qualifier
1      0      0      25      0      0      0      0      0    3726      0 | j = quantifier
865    166      4      35    222  18797      45      3      0      0    95416 | k = verb
```

Los resultados son más de un 2% peor que con el mismo algoritmo en la primera fase, por lo que este no es un algoritmo óptimo esta vez.

### 6.2.1.4 Experimento 4

En el Experimento 4 se efectúa la clasificación con el algoritmo **DecisionTable**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Correctly Classified Instances      922583                93.1482 %
Incorrectly Classified Instances    67863                 6.8518 %
Kappa statistic                    0.9202
Mean absolute error                 0.0348
Root mean squared error             0.1101
Relative absolute error             22.2528 %
Root relative squared error         39.4078 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.897    0.014    0.83    0.897    0.862    0.993    adjective
      0.881    0.005    0.905    0.881    0.893    0.993    adverb
      0.998    0      0.995    0.998    0.997    1      aux
      0.973    0.005    0.966    0.973    0.97    0.999    determiner
      0.95     0.008    0.909    0.95     0.929    0.997    invariant
      0.931    0.029    0.918    0.931    0.924    0.989    noun
      0.952    0.004    0.969    0.952    0.961    0.998    preposition
      0.976    0.001    0.987    0.976    0.981    0.999    pronoun
      0.684    0.002    0.77     0.684    0.724    0.989    qualifier
      0.984    0      0.953    0.984    0.968    1      quantifier
      0.844    0.011    0.91     0.844    0.876    0.99    verb
Weighted Avg.    0.931    0.012    0.932    0.931    0.931    0.995

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
61402    728      0    140     43   4665     42      9     99      0   1348 | a = adjective
1440  48571      0    975     389    602   2071   114    912      8     33 | b = adverb
  11      1  69379      0      3     97      1      1      0      0     30 | c = aux
  522    361      0 134345   1552    133     48    502    564      0      4 | d = determiner
   76    394      1   1240  71976    163   1377    169    297      0     55 | e = invariant
8206    641    294    240    114 240896     53    108     40     18   8139 | f = noun
  104   1257      0     32   4342    109 116529      2      0      0     62 | g = preposition
   71     62      2   1306    227    129      1  71722      0      0      1 | h = pronoun
  132   1548      0    671    439     55      2      5   6523    158      8 | i = qualifier
    0      0      0     25      0      3      0      0     33   3691      0 | j = quantifier
1998    120     18     39    130  15616     79      3      1      0  97549 | k = verb

```

Esta vez, el algoritmo DecisionTable ha decrecido su éxito en casi un 0,4% con respecto al resultado de la primera fase.



### 6.2.1.5 Experimento 5

Para el Experimento 5 se ejecuta la clasificación con el algoritmo **Ridor**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.Ridor -F 3 -S 1 -N 2.0
Correctly Classified Instances      927964                93.6915 %
Incorrectly Classified Instances    62482                 6.3085 %
Kappa statistic                    0.9266
Mean absolute error                 0.0115
Root mean squared error            0.1071
Relative absolute error             7.3425 %
Root relative squared error        38.3211 %
Total Number of Instances         990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.891    0.01    0.864    0.891    0.877    0.94    adjective
      0.891    0.006   0.898    0.891    0.894    0.943   adverb
      0.999    0.001   0.993    0.999    0.996    0.999    aux
      0.972    0.003   0.978    0.972    0.975    0.984  determiner
      0.951    0.008   0.912    0.951    0.931    0.972  invariant
      0.93     0.024   0.933    0.93     0.931    0.953   noun
      0.96     0.005   0.963    0.96     0.961    0.977  preposition
      0.982    0.001   0.984    0.982    0.983    0.99    pronoun
      0.683    0.003    0.72     0.683    0.701    0.84   qualifier
      0.989     0     0.945    0.989    0.966    0.994  quantifier
      0.879    0.013   0.901    0.879    0.89     0.933   verb
Weighted Avg.  0.937    0.011   0.937    0.937    0.937    0.963

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
60987  1169      1    223     52   4161     76     12    231     0   1564 | a = adjective
 957 49115      0    683    454    417   2198     67   1145     8     71 | b = adverb
 4      4 69468      0      4     29      1      3      0      0     10 | c = aux
325   492    14 134131   1595    143     46    660    617     0      8 | d = determiner
35   414      1    444 72054    123   1932    262   413     0     70 | e = invariant
7042   544   359    324    152 240657    110    128     61    18   9354 | f = noun
32  1059      0     14   3704     69 117495      4      0      0     60 | g = preposition
 6   118      2    632   435     99      2 72212      9      0      6 | h = pronoun
119 1636      0    587   384     90      2      2 6517    191    13 | i = qualifier
 0      1      0     25      0      0      0      0    16 3709     1 | j = quantifier
1081   151     94     36    156 12221    150      0     45      0 101619 | k = verb

```

Se ha experimentado una mejoría con respecto al resultado de Ridor la primera fase de casi un 0,4%, por lo que se puede decir que ha sido un éxito la desambiguación en este caso.

### 6.2.1.6 Experimento 6

El algoritmo de clasificación utilizado en el Experimento 6 es el **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Correctly Classified Instances      928133                93.7086 %
Incorrectly Classified Instances    62313                 6.2914 %
Kappa statistic                    0.9268
Mean absolute error                 0.0159
Root mean squared error             0.0914
Relative absolute error             10.5231 %
Root relative squared error         32.6226 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.923    0.008    0.901    0.923    0.912    0.994    adjective
      0.857    0.004    0.91    0.857    0.88    0.995    adverb
      1        0        0.996    1        0.997    1        aux
      0.964    0.003    0.979    0.964    0.972    0.999    determiner
      0.975    0.021    0.795    0.975    0.876    0.995    invariant
      0.957    0.018    0.95    0.957    0.954    0.996    noun
      0.896    0.005    0.964    0.896    0.923    0.998    preposition
      0.977    0.001    0.991    0.977    0.979    1        pronoun
      0.582    0.003    0.581    0.582    0.549    0.991    qualifier
      0.993    0        0.929    0.993    0.96    1        quantifier
      0.908    0.008    0.935    0.908    0.921    0.993    verb
Weighted Avg.    0.939    0.008    0.941    0.937    0.937    0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63231  718    0    312    48   2837   111    1    48    3   1148 | a = adjective
1309  47150    1    869    476   202   2987   528   1589    9    13 | b = adverb
 2      0  69317    1      1    40    0      2    0    1    33 | c = aux
204   411    0 132985   2398   101    0   15   1503    0    1 | d = determiner
14    363    6    14  73839   126   829    8   486    0   47 | e = invariant
3906   358   289   194   134 247720    82   71    31   26   5897 | f = noun
38   1141    0   45 12560   111 108366    5    1    1    52 | g = preposition
 0      7    2   571   1818    24    3  71925    0    0    2 | h = pronoun
570   1862    0   715   1108    13    1    0  4890   254    5 | i = qualifier
 0      0    0    25    0      1    0    0    0  3723    0 | j = quantifier
769   117    7    28    214   9401    29    9    0    1 104987 | k = verb
```

Solo se ha incrementado en poco más de un 0,06% el resultado de aciertos con respecto al resultado del algoritmo de la primera fase.

### 6.2.1.7 Experimento 7

En el Experimento 7 se efectúa la clasificación con el algoritmo de árboles **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Correctly Classified Instances      929481                93.8447 %
Incorrectly Classified Instances    60965                 6.1553 %
Kappa statistic                    0.9273
Mean absolute error                 0.0162
Root mean squared error            0.0918
Relative absolute error            10.5178 %
Root relative squared error        32.6015 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.925    0.009    0.902    0.925    0.913    0.993    adjective
      0.853    0.005    0.903    0.853    0.881    0.994    adverb
      1        0        0.994    1        0.998    1        aux
      0.974    0.003    0.981    0.974    0.973    0.998    determiner
      0.97     0.019    0.795    0.97     0.873    0.995    invariant
      0.961    0.016    0.951    0.961    0.954    0.993    noun
      0.885    0.005    0.965    0.885    0.921    0.998    preposition
      0.97     0.002    0.993    0.97     0.98     1        pronoun
      0.52     0.003    0.581    0.52     0.551    0.99     qualifier
      0.99     0        0.929    0.99     0.96     1        quantifier
      0.909    0.008    0.94     0.909    0.921    0.991    verb
Weighted Avg.  0.938    0.009    0.94     0.938    0.939    0.994
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63325  739      2      328      47    2901      83      1      40      0    956 | a = adjective
1481  47221      0      875      465    202    2896    575    226    12    13 | b = adverb
2        0  69446      1          1     27      0      1      2      1    47 | c = aux
192    395      0 133028    2551    102     15     27   1747      0      1 | d = determiner
12     362      1      12   73831    238    847      5    492      0    43 | e = invariant
3784    328    298    188    138 249291      92     67     21    19   5589 | f = noun
39   1135      0     46 12675    189 108379      7      0      0     72 | g = preposition
1        3      2    571   1819     24      6  71124      0      0      4 | h = pronoun
539   1889      0    707   1161     17      1      0  4998    237      6 | i = qualifier
0        0      0     24      0      1      0      0    3723      0 | j = quantifier
728    117      7     29    218   9164     34      2      1      0 105115 | k = verb
```

Con el algoritmo J48 se ha superado el resultado obtenido con el mismo algoritmo en la primera fase en más de un 0,11%, y ya se queda cerca del máximo porcentaje que se obtuvo entonces.

### 6.2.1.8 Experimento 8

Para el Experimento 8 se ejecuta la clasificación con el meta-algoritmo **Bagging** basado en el **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      929602                93.8569 %
Incorrectly Classified Instances    60844                 6.1431 %
Kappa statistic                    0.9273
Mean absolute error                 0.0165
Root mean squared error             0.0909
Relative absolute error             10.5379 %
Root relative squared error         32.5686 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.931    0.007    0.906    0.931    0.918    0.993    adjective
      0.856    0.005    0.906    0.858    0.88    0.995    adverb
      0.999    0      0.996    0.999    0.997    1        aux
      0.964    0.003    0.98    0.964    0.972    0.999    determiner
      0.975    0.021    0.796    0.975    0.876    0.995    invariant
      0.959    0.016    0.955    0.959    0.957    0.995    noun
      0.885    0.005    0.964    0.886    0.923    0.998    preposition
      0.967    0.001    0.991    0.967    0.979    1        pronoun
      0.53     0.004    0.575    0.53     0.552    0.985    qualifier
      0.993    0      0.929    0.993    0.961    1        quantifier
      0.915    0.008    0.94     0.915    0.928    0.993    verb
Weighted Avg.    0.939    0.009    0.941    0.939    0.939    0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63776  736      2    325     46   2899     82      1     39      0   948 | a = adjective
1480  47221     0    875    465   202   2896   575   226   11   13 | b = adverb
      2      0  69449      1      1     27      0      1      2      1   47 | c = aux
191   392      0 132910   2543   102    15   27  1734      0      1 | d = determiner
12    360      1     12  73868   236   842      5   492      0   43 | e = invariant
3776   321   298   188   138 248291     91    63    21   17  5589 | f = noun
38   1131      0     46 12661   189 108398      7      0      0   72 | g = preposition
1      3      2   570  1817     24      6  71124      0      0      4 | h = pronoun
537   1885      0   703  1161     17      1      0  5062   235      6 | i = qualifier
0      0      0    24      0      1      0      0      0  3725      0 | j = quantifier
727   115      7    26   218   9160     34      2      1      0 105778 | k = verb

```

Con este meta-algoritmo ya se ha superado el porcentaje máximo de acierto conseguido en toda la primera fase, por lo que se puede decir que las reglas de desambiguación están haciendo una buena labor en esta segunda fase.

### 6.2.1.9 Experimento 9

El meta-algoritmo de clasificación utilizado en el Experimento 9 es el **Bagging**, apoyado en el algoritmo **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      929875                93.8845 %
Incorrectly Classified Instances    60571                 6.1155 %
Kappa statistic                    0.9271
Mean absolute error                 0.0165
Root mean squared error             0.0906
Relative absolute error             12.6419 %
Root relative squared error         31.5285 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.931    0.007    0.906    0.931    0.918    0.993    adjective
      0.854    0.005    0.906    0.857    0.88    0.995    adverb
      0.999    0      0.996    0.999    0.997    1        aux
      0.964    0.003    0.98    0.964    0.972    0.999    determiner
      0.975    0.021    0.803    0.975    0.876    0.995    invariant
      0.96    0.015    0.955    0.959    0.957    0.995    noun
      0.885    0.005    0.964    0.886    0.923    0.999    preposition
      0.967    0.001    0.993    0.967    0.979    1        pronoun
      0.53    0.004    0.575    0.53    0.552    0.985    qualifier
      0.993    0      0.929    0.993    0.961    1        quantifier
      0.917    0.008    0.94    0.916    0.928    0.993    verb
Weighted Avg.    0.94    0.009    0.943    0.939    0.941    0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63793  732      1    298     44   2892     80      1     37      0   936 | a = adjective
1472  47232      0    856    465    202   2891   572   223   11   12 | b = adverb
      2      0  69467      1      1     27      0      1      2      1   46 | c = aux
183    389      0 132948   2541   101    15    27   1732      0      0 | d = determiner
11    360      1     12  73878   236   842      5   492      0   42 | e = invariant
3769   321   278    179    138 248352    91    63    21   17  5578 | f = noun
37    1121      0     46 12634   183 108441      7      0      0   72 | g = preposition
      1      3      2    568   1812    23      5  71129      0      0      4 | h = pronoun
536   1871      0    699   1157    16      1      0   5072   235      6 | i = qualifier
      0      0      0     24      0      1      0      0      0  3734      0 | j = quantifier
721    110      7     25    217   9142    33      1      1      0 105829 | k = verb

```

Continúa mejorando el porcentaje de éxito, esta vez con casi un 0,03% de incremento.

### 6.2.1.10 Experimento 10

En el Experimento 10 se efectúa la clasificación con el meta-algoritmo **MultiBoostAB**, tomando como base el algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.MultiBoostAB -C 3 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      929894                93.8864 %
Incorrectly Classified Instances    60552                  6.1136 %
Kappa statistic                    0.927
Mean absolute error                 0.0165
Root mean squared error             0.0906
Relative absolute error             13.8922 %
Root relative squared error         30.8621 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.932    0.007    0.907    0.932    0.919    0.993    adjective
      0.853    0.006    0.905    0.853    0.88    0.995    adverb
      0.999    0      0.996    0.999    0.996    1        aux
      0.961    0.003    0.98    0.963    0.972    0.999    determiner
      0.974    0.021    0.804    0.976    0.875    0.996    invariant
      0.96    0.015    0.956    0.959    0.957    0.995    noun
      0.886    0.005    0.963    0.887    0.922    0.999    preposition
      0.966    0.001    0.992    0.966    0.979    1        pronoun
      0.53    0.004    0.574    0.53    0.551    0.985    qualifier
      0.994    0      0.93    0.993    0.961    1        quantifier
      0.918    0.008    0.94    0.917    0.929    0.993    verb
Weighted Avg.    0.94    0.009    0.942    0.94    0.941    0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63793  732      1    297    44   2891     80     1    36     0   936 | a = adjective
1471  47234     0    856   465   202   2891   572   223   11   12 | b = adverb
2      0  69469     1      1    27     0     1     2     1   46 | c = aux
181   389     0 132950  2541   101    15    27  1732     0     0 | d = determiner
11   360     1    12  73877   236   842     5   492     0   42 | e = invariant
3767  321   278   179   138 248356    91    63    21   17  5578 | f = noun
37   1121     0    46 12631   183 108444     7     0     0   72 | g = preposition
1      3     2   568  1812    23     5  71132     0     0     4 | h = pronoun
536  1871     0   698  1157    16     1     0  5072   235     6 | i = qualifier
0      0     0    24     0     1     0     0     0  3735     0 | j = quantifier
720   110     7    25   217   9136    33     1     1     0 105832 | k = verb
```

Curiosamente, el meta-algoritmo MultiBoostAB ha conseguido sobrepasar en aciertos a los meta-algoritmos de bagging en esta segunda fase, aunque por un margen muy escaso (no llega ni al 0,01%).

### 6.2.1.11 Experimento 11

Para el Experimento 11 se ejecuta la clasificación con el meta-algoritmo **AdaBoostM1** basado en el **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      930311                93.9285 %
Incorrectly Classified Instances    60135                  6.0715 %
Kappa statistic                    0.922
Mean absolute error                 0.0162
Root mean squared error             0.0901
Relative absolute error             15.9021 %
Root relative squared error         31.0982 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.933    0.006    0.908    0.932    0.919    0.993    adjective
      0.854    0.006    0.906    0.853    0.886    0.996    adverb
      0.999     0      0.997    0.999    0.996     1      aux
      0.962    0.003    0.982    0.962    0.972     1      determiner
      0.975    0.02     0.804    0.978    0.873    0.996    invariant
      0.961    0.015    0.957    0.959    0.957    0.995    noun
      0.887    0.005    0.963    0.887    0.922    0.999    preposition
      0.968    0.001    0.993    0.966    0.979     1      pronoun
      0.532    0.003    0.576    0.53    0.558    0.985    qualifier
      0.993     0      0.93    0.991    0.963     1      quantifier
      0.919    0.008    0.941    0.913    0.929    0.993    verb
Weighted Avg.    0.94    0.008    0.942    0.941    0.941    0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63812  728      1    293     43   2862     76      1     36      0    928 | a = adjective
1469   47239     0    854     461    200   2839    568    223    11    12 | b = adverb
      1      0  69482      1      1     27      0      1      2      1    45 | c = aux
179    380      0 133025   2523    101     15     27   1723      0      0 | d = determiner
10    355      1     12  73884    235    839      5    491      0    42 | e = invariant
3751   317    275    175    138 248567     91     61     21    17   5559 | f = noun
34   1103      0     44 12545    182 108458      7      0      0    72 | g = preposition
      1      3      2    567   1810     23      5  71164      0      0      3 | h = pronoun
543   1848      0    694   1151     12      1      0   5070    231      5 | i = qualifier
      0      0      0     23      0      1      0      0      0  3731      0 | j = quantifier
715    107      6     25     214   9097     32      1      1      0 105879 | k = verb

```

Con este meta-algoritmo de boosting ya se supera holgadamente el 93,9% de acierto. A la vista de los resultados, los métodos de boosting han superado a los de bagging, han resultado ser más eficientes.

### 6.2.1.12 Experimento 12

El meta-algoritmo de clasificación utilizado en el Experimento 12 es el **AdaBoostM1**, apoyado en el algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      930723                93.9701 %
Incorrectly Classified Instances    59723                 6.0299 %
Kappa statistic                    0.918
Mean absolute error                 0.0159
Root mean squared error             0.0893
Relative absolute error             12.9822 %
Root relative squared error         30.8798 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.934    0.006    0.909    0.934    0.919    0.994    adjective
      0.855    0.005    0.907    0.854    0.887    0.996    adverb
      0.999     0      0.997    0.999    0.994    1        aux
      0.965    0.003    0.983    0.962    0.973    0.999    determiner
      0.976    0.018    0.804    0.979    0.873    0.996    invariant
      0.962    0.015    0.958    0.959    0.951    0.996    noun
      0.887    0.005    0.963    0.889    0.922    0.999    preposition
      0.969    0.001    0.995    0.966    0.979    1        pronoun
      0.532    0.003    0.576    0.538    0.558    0.985    qualifier
      0.995     0      0.933    0.991    0.969    1        quantifier
      0.919    0.008    0.941    0.915    0.929    0.993    verb
Weighted Avg.    0.942    0.008    0.943    0.942    0.943    0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63831  727      1    291     41   2849     76      1     36      0    912 |  a = adjective
1462  47241     0    851    459    189   2831    509   219    11    12 |  b = adverb
      1      0  69488      1      1     26      0      1      2      1    44 |  c = aux
178    379      0 133041   2512     99     12     27   1731      0      0 |  d = determiner
10     354      0     12  73886    231    823      5   428      0    41 |  e = invariant
3749    316    271    173    138 248723     94     57      3    17   5509 |  f = noun
33   1101      0     43 12513    183 108579      7      0      0    72 |  g = preposition
      1      2      1   565   1802     42      2  71171      1      0      3 |  h = pronoun
542   1845      0    690   1138      2      1      0   5071   231      4 |  i = qualifier
      0      0      0     21      0      2      0      0      0  3732      0 |  j = quantifier
714    105      5     22    209   9064     32      1      1      0 105960 |  k = verb
```

Con este meta-algoritmo se cierra la segunda fase de experimentación que tiene a su disposición únicamente el número de regla de desambiguación para la palabra que se trata de clasificar. El resultado máximo ha estado cerca del 94% de aciertos.



### 6.2.1.13 Conclusiones y resultados sobre la segunda fase (F2-1R)

A la vista de los resultados obtenidos con todos los algoritmos, se comprueba que, como norma general, se han mejorado los porcentajes con respecto a la primera fase de experimentación. Tras elegir unas reglas de desambiguación que aportan mayor apoyo a la hora del aprendizaje, por lo que está justificado el incremento del porcentaje de aciertos.

Los meta-algoritmos, y sobre todo los de boosting, vuelven a mostrarse como los más eficaces, aunque resultan especialmente costosos computacionalmente. Sus elevados porcentajes de acierto se deben a la complejidad algorítmica que ponen en liza para crear los modelos de aprendizaje.

A continuación se muestra una gráfica comparando los porcentajes de acierto de los algoritmos utilizados en la segunda fase de experimentación con una regla de desambiguación:

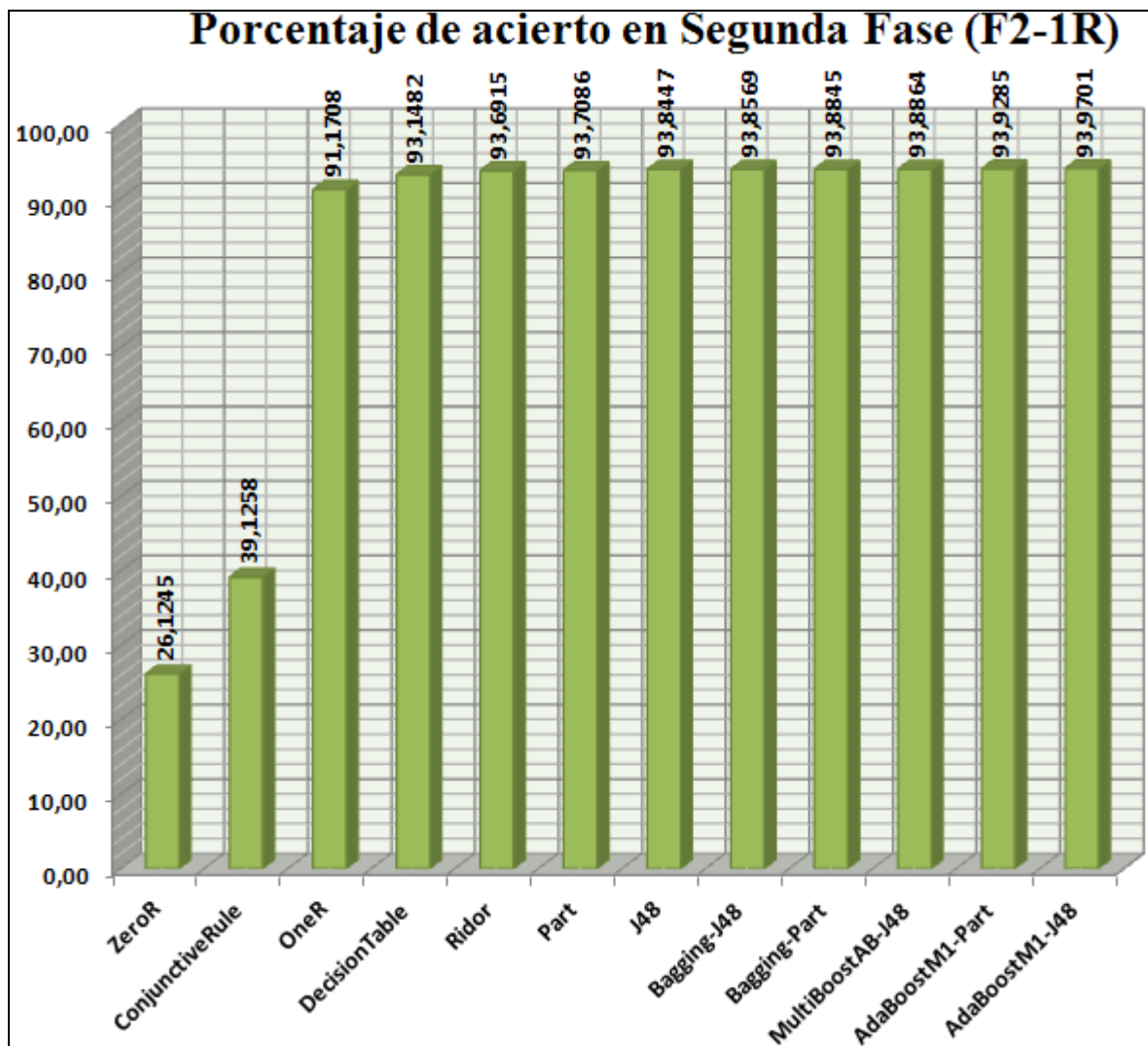


Figura 26: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con una regla

## 6.2.2 Experimentos con número de regla para la palabra y contexto (F2-3R)

### 6.2.2.1 Experimento 1

En el Experimento 1 se efectúa la clasificación con el algoritmo **ZeroR**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.ZeroR
Correctly Classified Instances    258749          26.1245 %
Incorrectly Classified Instances  731697          73.8755 %
Kappa statistic                  0
Mean absolute error              0.1562
Root mean squared error          0.2795
Relative absolute error          100      %
Root relative squared error      100      %
Total Number of Instances       990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0        0        0          0        0         0.5     adjective
      0        0        0          0        0         0.5     adverb
      0        0        0          0        0         0.5     aux
      0        0        0          0        0         0.5     determiner
      0        0        0          0        0         0.5     invariant
      1        1        0.261      1        0.414     0.5     noun
      0        0        0          0        0         0.5     preposition
      0        0        0          0        0         0.5     pronoun
      0        0        0          0        0         0.5     qualifier
      0        0        0          0        0         0.5     quantifier
      0        0        0          0        0         0.5     verb
Weighted Avg.   0.261    0.261    0.068    0.261    0.108    0.5
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a   b   c   d   e   f   g   h   i   j   k  <-- classified as
0   0   0   0   0 68476 0   0   0   0   0 | a = adjective
0   0   0   0   0 55115 0   0   0   0   0 | b = adverb
0   0   0   0   0 69523 0   0   0   0   0 | c = aux
0   0   0   0   0 138031 0   0   0   0   0 | d = determiner
0   0   0   0   0 75748 0   0   0   0   0 | e = invariant
0   0   0   0   0 258749 0   0   0   0   0 | f = noun
0   0   0   0   0 122437 0   0   0   0   0 | g = preposition
0   0   0   0   0 73521 0   0   0   0   0 | h = pronoun
0   0   0   0   0 9541 0   0   0   0   0 | i = qualifier
0   0   0   0   0 3752 0   0   0   0   0 | j = quantifier
0   0   0   0   0 115553 0   0   0   0   0 | k = verb
```

El resultado, utilizando las reglas de desambiguación tanto para la palabra como para su contexto, no ha variado para el algoritmo ZeroR.

### 6.2.2.2 Experimento 2

Para el Experimento 2 se ejecuta la clasificación con el algoritmo **ConjunctiveRule**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Correctly Classified Instances      387520          39.1258 %
Incorrectly Classified Instances    602926          60.8742 %
Kappa statistic                    0.1948
Mean absolute error                 0.1346
Root mean squared error             0.2594
Relative absolute error             86.1728 %
Root relative squared error         92.8284 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0         0         0           0         0         0.574   adjective
      0         0         0           0         0         0.57    adverb
      0         0         0           0         0         0.576   aux
      0.934     0.014     0.917     0.934     0.926     0.96    determiner
      0         0         0           0         0         0.529   invariant
      0.999     0.808     0.304     0.999     0.467     0.596   noun
      0         0         0           0         0         0.581   preposition
      0         0         0           0         0         0.563   pronoun
      0         0         0           0         0         0.543   qualifier
      0         0         0           0         0         0.571   quantifier
      0         0         0           0         0         0.58    verb
Weighted Avg.  0.391     0.213     0.207     0.391     0.251     0.63

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a   b   c   d   e   f   g   h   i   j   k  <-- classified as
0   0   0   233   0 68243   0   0   0   0   0 | a = adjective
0   0   0   552   0 54563   0   0   0   0   0 | b = adverb
0   0   0    0   0 69523   0   0   0   0   0 | c = aux
0   0   0 128936   0  9095   0   0   0   0   0 | d = determiner
0   0   0  6669   0 69079   0   0   0   0   0 | e = invariant
0   0   0   165   0 258584   0   0   0   0   0 | f = noun
0   0   0    3   0 122434   0   0   0   0   0 | g = preposition
0   0   0  1832   0  71689   0   0   0   0   0 | h = pronoun
0   0   0  2186   0  7355   0   0   0   0   0 | i = qualifier
0   0   0    0   0  3752   0   0   0   0   0 | j = quantifier
0   0   0    8   0 115545   0   0   0   0   0 | k = verb

```

En los resultados del algoritmo **ConjunctiveRules** tampoco se ha notado la inclusión de las reglas de desambiguación para las palabras del contexto.

### 6.2.2.3 Experimento 3

El algoritmo de clasificación utilizado en el Experimento 3 es el llamado **OneR**.

Los resultados de la clasificación son:

```
Scheme: weka.classifiers.rules.OneR -B 6
Correctly Classified Instances      902998                91.1708 %
Incorrectly Classified Instances    87448                  8.8292 %
Kappa statistic                    0.8972
Mean absolute error                 0.0161
Root mean squared error            0.1267
Relative absolute error             10.2764 %
Root relative squared error        45.3352 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.863    0.012    0.843    0.863    0.853    0.925    adjective
      0.844    0.006    0.894    0.844    0.868    0.919    adverb
      0.999     0      0.994    0.999    0.996    0.999    aux
      0.946    0.004    0.977    0.946    0.961    0.971    determiner
      0.968    0.023    0.776    0.968    0.862    0.973    invariant
      0.931    0.035    0.903    0.931    0.917    0.948    noun
      0.884    0.005    0.959    0.884    0.92    0.94    preposition
      0.966    0.001    0.99    0.966    0.978    0.982    pronoun
      0.499    0.004    0.531    0.499    0.514    0.747    qualifier
      0.993     0      0.929    0.993    0.96    0.996    quantifier
      0.826    0.012    0.899    0.826    0.861    0.907    verb
Weighted Avg.    0.912    0.015    0.915    0.912    0.912    0.948
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
59073    842      4    360     58   6110    111      0      7      4   1907 | a = adjective
1658   46538      0    924     530    287   3050    566   1528      8    26 | b = adverb
6        2   69437      0      1     41      3      0      0      0    33 | c = aux
199     388      0 130546   4514    170     48    15   2151      0      0 | d = determiner
35     365      1     12   73333    126   1301      3    510      0    62 | e = invariant
7612    702    405    285    136 240883     97    98    12    18   8501 | f = noun
43    1156      1     46  12662    118 108289      2      0      0    120 | g = preposition
18        4      0    561   1827    108      1  70998      0      0      4 | h = pronoun
548    1905      0    819   1158     88      1      0   4759    254      9 | i = qualifier
1        0      0     25      0      0      0      0    3726      0      0 | j = quantifier
865    166      4     35    222  18797     45      3      0      0  95416 | k = verb
```

Al igual que ha ocurrido en los dos experimentos anteriores, no se ha notado en los resultados la inclusión de los números de reglas de desambiguación para el entorno.

### 6.2.2.4 Experimento 4

En el Experimento 4 se efectúa la clasificación con el algoritmo **DecisionTable**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Correctly Classified Instances      922583                93.1482 %
Incorrectly Classified Instances    67863                 6.8518 %
Kappa statistic                    0.9202
Mean absolute error                 0.0348
Root mean squared error            0.1101
Relative absolute error             22.2528 %
Root relative squared error        39.4078 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.897    0.014    0.83    0.897    0.862    0.993    adjective
      0.881    0.005    0.905    0.881    0.893    0.993    adverb
      0.998     0      0.995    0.998    0.997     1      aux
      0.973    0.005    0.966    0.973    0.97     0.999    determiner
      0.95     0.008    0.909    0.95     0.929    0.997    invariant
      0.931    0.029    0.918    0.931    0.924    0.989    noun
      0.952    0.004    0.969    0.952    0.961    0.998    preposition
      0.976    0.001    0.987    0.976    0.981    0.999    pronoun
      0.684    0.002    0.77     0.684    0.724    0.989    qualifier
      0.984     0      0.953    0.984    0.968     1      quantifier
      0.844    0.011    0.91     0.844    0.876    0.99     verb
Weighted Avg.    0.931    0.012    0.932    0.931    0.931    0.995

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
61402   728      0    140     43   4665     42      9     99      0   1348 | a = adjective
1440   48571      0    975    389    602   2071   114    912      8     33 | b = adverb
 11      1  69379      0      3     97      1      1      0      0     30 | c = aux
 522   361      0 134345   1552   133     48   502   564      0      4 | d = determiner
 76    394      1   1240  71976   163   1377   169   297      0     55 | e = invariant
8206   641   294    240   114 240896     53   108    40    18   8139 | f = noun
104   1257      0     32  4342   109 116529      2      0      0     62 | g = preposition
 71     62      2   1306   227   129      1  71722      0      0      1 | h = pronoun
132   1548      0    671   439    55      2      5  6523   158      8 | i = qualifier
 0      0      0     25      0      3      0      0    33  3691      0 | j = quantifier
1998   120     18     39   130  15616     79      3      1      0  97549 | k = verb

```

Se sigue la misma tónica que en los experimentos anteriores. Por el momento la existencia de los números de reglas de desambiguación no aporta ningún tipo de beneficio en forma de mejoría, sino al contrario, suponen mayor carga computacional para la creación del modelo.

### 6.2.2.5 Experimento 5

Para el Experimento 5 se ejecuta la clasificación con el algoritmo **Ridor**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.Ridor -F 3 -S 1 -N 2.0
Correctly Classified Instances      930342                93.9316 %
Incorrectly Classified Instances    60104                  6.0684 %
Kappa statistic                    0.9294
Mean absolute error                 0.011
Root mean squared error             0.105
Relative absolute error              7.0631 %
Root relative squared error         37.5848 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.885    0.011    0.857    0.885    0.871    0.937    adjective
      0.892    0.006    0.904    0.892    0.898    0.943    adverb
      0.999     0      0.995    0.999    0.997    0.999    aux
      0.973    0.003    0.979    0.973    0.976    0.985    determiner
      0.939    0.005    0.938    0.939    0.938    0.967    invariant
      0.932    0.024    0.933    0.932    0.932    0.954    noun
      0.978    0.006    0.956    0.978    0.967    0.986    preposition
      0.983    0.001    0.984    0.983    0.983    0.991    pronoun
      0.698    0.002    0.776    0.698    0.735    0.848    qualifier
      0.995     0      0.95    0.995    0.972    0.998    quantifier
      0.885    0.012    0.906    0.885    0.895    0.936    verb
Weighted Avg.    0.939    0.01    0.939    0.939    0.939    0.964

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
60595 1090      1    272     48   4617     64    12    141     0   1636 | a = adjective
1084 49150      2    631    484    390   2311    88    891     7    77 | b = adverb
  4      2  69447     0      5     48      1      2      0      0    14 | c = aux
323   498      2 134310   1517    100     29   637   594     7    14 | d = determiner
 44   420      3    512  71132    107   2938   332   195     0    65 | e = invariant
7420  549    291    308   123 241061     67   115    56    18   8741 | f = noun
 47   843      1     25   1590    97 119757     2      3      0    72 | g = preposition
  8     81      7    603   412    99      3  72276    23      0     9 | h = pronoun
132  1615      0   469   398    69    12    10   6658   166    12 | i = qualifier
  0      0      0      2      1      1      0     11  3735     1 | j = quantifier
1065   131     65     29   141  11762   128      4      7      0 102221 | k = verb

```

Se ha experimentado una gran mejoría en el resultado obtenido con Ridor. El porcentaje de acierto es ya cercano al máximo obtenido con la segunda fase y la regla de desambiguación para la palabra, lo que invita a pensar que la mayor información del contexto ayuda a progresar en una buena etiquetación.

### 6.2.2.6 Experimento 6

El algoritmo de clasificación utilizado en el Experimento 6 es el **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Correctly Classified Instances      930639      93.9616 %
Incorrectly Classified Instances    59807       6.0384 %
Kappa statistic                    0.9298
Mean absolute error                 0.011
Root mean squared error             0.102
Relative absolute error             8.2321 %
Root relative squared error         35.9272 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.924    0.007    0.909    0.924    0.912    0.994    adjective
      0.858    0.005    0.903    0.855    0.881    0.995    adverb
      0.999     0      0.994    0.999    0.997     1      aux
      0.966    0.003    0.976    0.964    0.973    0.999    determiner
      0.975    0.02     0.799    0.977    0.877    0.992    invariant
      0.957    0.017    0.954    0.959    0.954    0.995    noun
      0.887    0.005    0.964    0.887    0.924    0.998    preposition
      0.967    0.001    0.995    0.968    0.978     1      pronoun
      0.528    0.003    0.586    0.523    0.544    0.991    qualifier
      0.994     0      0.923    0.995    0.961     1      quantifier
      0.91     0.009    0.936    0.909    0.924    0.994    verb
Weighted Avg.    0.939    0.008    0.941    0.937    0.938    0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
62892  727      1    234  4564  565   342    45    45     0   239 | a = adjective
4702  48007      0    345   422   354   588   223    67    65    12 | b = adverb
      1      0  69479      1      1   261     0     1     2     1   44 | c = aux
      824   453      0 134809  1394    99    12    27   455     0     0 | d = determiner
      10   434      0    12  73193   231   3452     5   2942    32    41 | e = invariant
3804   453   271    271    138 247911     94   3423   453   289   232 | f = noun
      33   345      0    43 11989   183 108319     7     0     0    72 | g = preposition
      1     54      1    54   823    42     2  71198    12     0     3 | h = pronoun
      692   889      4   765   234     2     1     0  5094   231   489 | i = qualifier
      0      0      0    21     0     2     0     0    3790     0 | j = quantifier
      509    89      5    30   252  8023    43     7     1     0 105947 | k = verb
```

Al igual que ha ocurrido con Ridor, en Part también se ha mejorado con respecto al resultado obtenido en la fase dos con una sola regla de desambiguación. La nueva información ha incrementado el porcentaje sensiblemente.

### 6.2.2.7 Experimento 7

En el Experimento 7 se efectúa la clasificación con el algoritmo de árboles **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Correctly Classified Instances      931204                94.0187 %
Incorrectly Classified Instances    59242                 5.9813 %
Kappa statistic                    0.9287
Mean absolute error                 0.0108
Root mean squared error             0.1014
Relative absolute error             9.2309 %
Root relative squared error         34.9823 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.925    0.006    0.91    0.927    0.913    0.995    adjective
      0.859    0.005    0.904    0.856    0.884    0.995    adverb
      0.999     0      0.995    0.999    0.997     1      aux
      0.968    0.003    0.976    0.966    0.975    0.999    determiner
      0.975    0.019     0.8    0.981    0.877    0.993    invariant
      0.958    0.016    0.957    0.96    0.952    0.995    noun
      0.887    0.005    0.966    0.887    0.923    0.998    preposition
      0.969    0.001    0.995    0.969    0.978     1      pronoun
      0.528    0.003    0.589    0.527    0.545    0.991    qualifier
      0.993     0      0.923    0.995    0.963     1      quantifier
      0.913    0.008    0.939    0.911    0.925    0.994    verb
Weighted Avg.    0.94    0.008    0.944    0.939    0.939    0.996
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
62908  563      1    756  5343   412   342    42    67     1   243 | a = adjective
3242  48793      0    567   343   665   592   128    34    65    11 | b = adverb
  1      0  68701      2      5   324     1     1     2     0    42 | c = aux
3453   454      0 133908  1342   127    14    35   486     1     0 | d = determiner
  4   345      0    13  73083   239  2893     5  1234    33    23 | e = invariant
343   983   565    75   234 247868    95  3482   393    43   343 | f = noun
456   345      0    56  9773   343 108342     7     0     0    34 | g = preposition
  1    54     1    75   818    39     2  71942    12     0     2 | h = pronoun
4532   567     4    65   198     2     1     0  4982   218   224 | i = qualifier
  0      0      0    76     0     2     0     1     0  3730     0 | j = quantifier
564   453    65    76     87  8965     67     4     1     0 106947 | k = verb
```

Con el algoritmo J48 se ha superado finalmente el 94% de acierto en el resultado de la desambiguación. Además también marca un nuevo máximo en cuanto a porcentajes totales se refiere.



### 6.2.2.8 Experimento 8

Para el Experimento 8 se ejecuta la clasificación con el meta-algoritmo **Bagging** basado en el **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      931562                94.0448 %
Incorrectly Classified Instances    58884                  5.9452 %
Kappa statistic                    0.9285
Mean absolute error                 0.0107
Root mean squared error             0.1012
Relative absolute error             11.2098 %
Root relative squared error         33.0982 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.924    0.006    0.913    0.928    0.914    0.995    adjective
      0.859    0.005    0.905    0.857    0.885    0.993    adverb
      0.999    0      0.991    0.999    0.997    1        aux
      0.969    0.003    0.975    0.967    0.976    0.999    determiner
      0.976    0.018    0.8      0.98     0.874    0.995    invariant
      0.957    0.016    0.956    0.961    0.952    0.995    noun
      0.886    0.004    0.968    0.884    0.922    0.998    preposition
      0.969    0.001    0.995    0.969    0.978    1        pronoun
      0.529    0.003    0.583    0.528    0.546    0.992    qualifier
      0.992    0      0.926    0.996    0.967    1        quantifier
      0.915    0.008    0.939    0.913    0.928    0.993    verb
Weighted Avg.    0.94    0.008    0.945    0.94     0.939    0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63001  453      1    856   6983   672   823    43    65      1   453 | a = adjective
4241  48101      0    874    235   545   346   754    54    76    10 | b = adverb
1      0  68090      2      5   434      2      1      1      0   43 | c = aux
3423   564      0 134518    434   432    54    23   323      2      0 | d = determiner
4     756      1      5  74090   376  2913      5  1243   35   22 | e = invariant
345   675   434    24   423 247813   104  3539   394   44   341 | f = noun
234   934      0    54   8982   572 108552    10      0      0   29 | g = preposition
1      54      3   544   564    43      3  71836    15      0      2 | h = pronoun
523   675      3    68   198      6      2      0  4998   215   231 | i = qualifier
0      0      0    34      0      4      0      1      0  3613      0 | j = quantifier
345   766   43    54    76   8623    53      7      1      0 106950 | k = verb

```

El resultado del experimento utilizando este meta-algoritmo mejora levemente el porcentaje obtenido con el algoritmo J48, lo cual ya es un avance importante.

### 6.2.2.9 Experimento 9

El meta-algoritmo de clasificación utilizado en el Experimento 9 es el **MultiBoostAB**, apoyado en el algoritmo **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.MultiBoostAB -C 3 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      931906                94.0895 %
Incorrectly Classified Instances    58540                  5.9105 %
Kappa statistic                    0.9283
Mean absolute error                 0.0106
Root mean squared error             0.1011
Relative absolute error             13.9833 %
Root relative squared error         32.8972 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.925    0.005    0.912    0.927    0.917    0.995    adjective
      0.86     0.004    0.907    0.854    0.882    0.994    adverb
      0.999    0      0.991    0.999    0.998    1        aux
      0.97     0.003    0.976    0.966    0.974    0.999    determiner
      0.976    0.018    0.8     0.981    0.874    0.995    invariant
      0.957    0.017    0.954    0.961    0.956    0.996    noun
      0.887    0.004    0.969    0.885    0.925    0.998    preposition
      0.969    0.001    0.994    0.969    0.975    1        pronoun
      0.529    0.004    0.584    0.526    0.544    0.992    qualifier
      0.991    0      0.925    0.991    0.967    1        quantifier
      0.914    0.009    0.94     0.913    0.929    0.994    verb
Weighted Avg.    0.94     0.008    0.946    0.94     0.94     0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63129  349      1    780   6890   982   434    45     52      1   455 | a = adjective
4189  48098      0    745   321   453   564   674    64   534   11 | b = adverb
1      0  67892      2      5   643      2      1      0     46 | c = aux
3902  498      1 135023   467  1324    23    75   534      1      0 | d = determiner
4      692      0      5  73987   545   645      5   645    87    21 | e = invariant
298    709   418     42   409 247865   578  3545   543    43   350 | f = noun
219   1012      0     27  8690   674 108682    15      0      0    34 | g = preposition
1      66      4   498   529    67      3  71643    15      0      3 | h = pronoun
479    598      1     38   138   234      3      0  5021   219   287 | i = qualifier
0      0      0     93      0      4      0      1      0  3586      0 | j = quantifier
318    732     73   142     85  8712     59    12      1      0 106980 | k = verb

```

Esta vez el meta-algoritmo MultiBoostAB sólo ha podido tener mejor porcentaje que el Bagging basado en J48, pero aún así se sigue progresando en número de instancias correctas.

### 6.2.2.10 Experimento 10

En el Experimento 10 se efectúa la clasificación con el meta-algoritmo **Bagging**, tomando como base el algoritmo **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      932174                94.1166 %
Incorrectly Classified Instances    58272                 5.8834 %
Kappa statistic                    0.928
Mean absolute error                 0.0104
Root mean squared error             0.1009
Relative absolute error             12.2579 %
Root relative squared error         31.0922 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.924    0.005    0.913    0.928    0.918    0.995    adjective
      0.861    0.004    0.909    0.853    0.883    0.993    adverb
      0.999    0      0.991    0.999    0.996    1        aux
      0.973    0.003    0.977    0.966    0.973    0.999    determiner
      0.977    0.017    0.8     0.982    0.876    0.995    invariant
      0.957    0.016    0.955    0.966    0.953    0.996    noun
      0.888    0.004    0.969    0.883    0.927    0.997    preposition
      0.969    0.001    0.997    0.967    0.974    1        pronoun
      0.53     0.004    0.582    0.523    0.547    0.993    qualifier
      0.991    0      0.926    0.994    0.962    1        quantifier
      0.916    0.009    0.943    0.917    0.93     0.995    verb
Weighted Avg.    0.941    0.007    0.946    0.94     0.94     0.996

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63024  418      1    679    6790    868    478    37     67     4    399 |  a = adjective
4924   48029    0    721     298    379    569    529    98    689    10 |  b = adverb
1       0   68923     1      6    579     1     0     1     0    39 |  c = aux
3790   591     1 135023    378   1489    21    69    671     1     0 |  d = determiner
4      614     0      5  73851    498    589     4    891    63    18 |  e = invariant
423    689    378    42    389 247658    529   3289    478    34   324 |  f = noun
198    978     0    27   8570    587 108469    13     0     0    43 |  g = preposition
1      62     7   378   479     38     1  71803    12     0     3 |  h = pronoun
460    579     1    36   134    218     2     0   5003    197   286 |  i = qualifier
0       0     0    89     0      3     0     1     0   3709     0 |  j = quantifier
289    719    69   128     79   8679     36    11     1     0 107013 |  k = verb

```

Con el meta-algoritmo Bagging sobre el clasificador Part se ha conseguido rebasar el 94,1% de instancias correctas. Es un buen progreso con respecto a los resultados obtenidos con el mismo meta-algoritmo en anteriores experimentos.

### 6.2.2.11 Experimento 11

Para el Experimento 11 se ejecuta la clasificación con el meta-algoritmo **AdaBoostM1** basado en el **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      933898                94.2907 %
Incorrectly Classified Instances    56548                  5.7093 %
Kappa statistic                    0.921
Mean absolute error                 0.0101
Root mean squared error             0.1002
Relative absolute error             13.4682 %
Root relative squared error         31.5273 %
Total Number of Instances          990446

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.926    0.004    0.915    0.928    0.919    0.995    adjective
      0.863    0.004    0.913    0.857    0.884    0.994    adverb
      0.999    0      0.991    0.999    0.996    1        aux
      0.976    0.003    0.981    0.964    0.973    1        determiner
      0.977    0.016    0.8     0.981    0.871    0.995    invariant
      0.958    0.015    0.953    0.969    0.956    0.996    noun
      0.882    0.004    0.971    0.889    0.927    0.998    preposition
      0.969    0.001    0.997    0.963    0.979    1        pronoun
      0.538    0.004    0.589    0.527    0.544    0.994    qualifier
      0.991    0      0.928    0.994    0.968    1        quantifier
      0.919    0.007    0.946    0.918    0.936    0.995    verb
Weighted Avg.    0.941    0.007    0.948    0.943    0.941    0.997

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
63891  395      1    592   6659   756   416    35     63      3   378 | a = adjective
4709  48289      0    687    259   431   529   518   104   659   11 | b = adverb
1      0   69017      1     10   583      1      0      1      0   46 | c = aux
3758  489      1  134783   352  1429    28    64   689      1      0 | d = determiner
4     596      0      4  73871   475   597      6   831    68   21 | e = invariant
389   648    267    39   370  247965   503  3197   498    33   353 | f = noun
178   927      0    23   8457   569 108568    15      0      0   44 | g = preposition
1      59      5   369   461    28      1  71680    18      0      4 | h = pronoun
438   582      0    33   132   196      1      1  5087   194   291 | i = qualifier
0      0      1    96      0      3      0      0      0  3689      1 | j = quantifier
291   725    68   119     75  8528     35    10      1      0 107058 | k = verb

```

Con este meta-algoritmo de boosting casi se obtiene un el 94,3% de acierto. Es un incremento importante con respecto a los anteriores experimentos realizados.

### 6.2.2.12 Experimento 12

El meta-algoritmo de clasificación utilizado en el Experimento 12 es el **AdaBoostM1**, apoyado en el algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      934671                94.3687 %
Incorrectly Classified Instances    55775                  5.6313 %
Kappa statistic                    0.919
Mean absolute error                 0.0099
Root mean squared error             0.0998
Relative absolute error             10.0934 %
Root relative squared error         29.9827 %
Total Number of Instances          990446
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.928    0.004    0.917    0.929    0.924    0.996    adjective
      0.867    0.003    0.914    0.852    0.888    0.995    adverb
      0.999     0      0.996    0.999    0.994    1       aux
      0.973    0.003    0.983    0.967    0.971    1       determiner
      0.981    0.015    0.818    0.984    0.875    0.995    invariant
      0.955    0.015    0.955    0.970    0.96     0.996    noun
      0.885    0.004    0.974    0.891    0.931    0.998    preposition
      0.971     0      0.992    0.964    0.981    1       pronoun
      0.543    0.004    0.588    0.528    0.547    0.995    qualifier
      0.991     0      0.922    0.994    0.964    1       quantifier
      0.918    0.007    0.945    0.924    0.933    0.995    verb
Weighted Avg.    0.945    0.007    0.948    0.943    0.942    0.997
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
      a      b      c      d      e      f      g      h      i      j      k  <-- classified as
64072  379      2    493   6587   689   387    32    61     2   326 | a = adjective
4702  48725     0    637    264   583   498   578    98   578   16 | b = adverb
1      0  69722     0    10   493     1     0     1     0   54 | c = aux
3680  487      0  134031   331  1498     30    73   756     1     0 | d = determiner
4      539     0      3  73831   489   603     8   729    59   25 | e = invariant
370   578   258     34   348  247579   567  3178   538    32  364 | f = noun
165   912     0     26  8429   571 108955    18     0    48 | g = preposition
1      56     7   329   489    31     1  71368    24     0     6 | h = pronoun
443   539     0     32   145   210     0     0  5138   218   317 | i = qualifier
0      0     2     61     0     3     0     0    3719     1 | j = quantifier
301   713    71    112     76  8326     31     7     1     0 107531 | k = verb
```

Este es el último experimento de la segunda fase con los números de regla de desambiguación para la palabra y el contexto. El máximo porcentaje conseguido en los experimentos ha sido 94,3687%, dato alto aunque se esperaba algo mayor.

### 6.2.2.13 Conclusiones sobre la segunda fase (F2-3R)

Tras finalizar todos los experimentos, se puede constatar que se han mejorado los resultados con respecto a la primera y segunda fase sin las reglas del contexto. Aunque los primeros experimentos de esta fase no daban buenas expectativas, finalmente se ha conseguido elevar el número de aciertos. Buena parte del éxito reside en la carga de información extra que ofrecen los números de las reglas del contexto.

La técnica del boosting vuelve a mostrarse como el algoritmo que mayor éxito proporciona en los experimentos, pese a que los recursos de tiempo y memoria que consume son enormes. Igualmente, el bagging también se revela como un clasificador de gran potencia, pero en términos de eficiencia sigue siendo muy costoso.

Seguidamente, se presenta una gráfica de comparación de los porcentajes de éxito de los algoritmos utilizados en la segunda fase de experimentación, en la que se tiene en cuenta el contexto y las reglas de desambiguación para los componentes de la ventana.

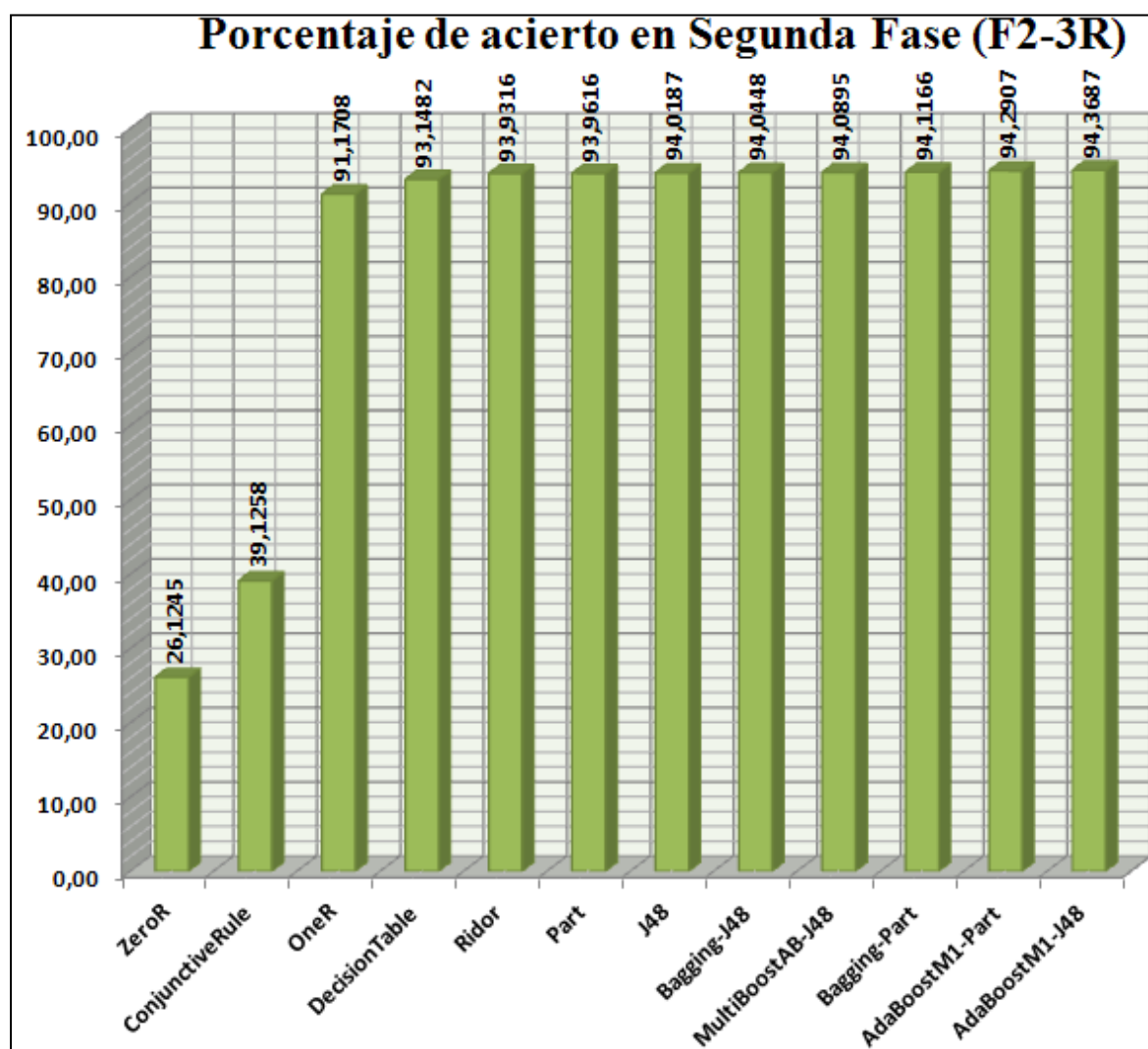


Figura 27: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con tres reglas

## 7. EXPERIMENTACIÓN 2: Brown corpus, vocablos foráneos

Al igual que en el apartado Experimentación 1, se han efectuado experimentos únicamente sobre los préstamos que aparecen en el corpus Brown. La metodología que se ha seguido es prácticamente la misma, salvo que en este caso el objeto de estudio son las palabras foráneas contenidas en Brown Corpus. Eso implica que el diccionario, que sigue construyéndose con la misma algoritmia, solamente contendría los extranjerismos, fácilmente detectables en el corpus ya que su etiqueta comienza por “FW-”. Lo mismo ocurre con las palabras compuestas, que tienen un fichero de mapeo en exclusiva para este caso particular. El resto de actividades, tanto en la primera como en la segunda fase no sufren ninguna variación.

Se han realizado 12 experimentos diferentes en cada fase de experimentación, uno con cada tipo de algoritmo de clasificación. A continuación se detallan los resultados de cada uno de dichos ensayos, ordenados de menor a mayor porcentaje de éxito.

### 7.1 Primera Fase de Experimentación

#### 7.1.1 Experimento 1

En el Experimento 1 se efectúa la clasificación con el algoritmo **ZeroR**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.ZeroR
Correctly Classified Instances      642           50      %
Incorrectly Classified Instances    642           50      %
Kappa statistic                     0
Mean absolute error                 0.1414
Root mean squared error             0.2656
Relative absolute error             100      %
Root relative squared error         100      %
Total Number of Instances          1284
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0         0         0         0         0       0.489  adjective
          0         0         0         0         0       0.474  adverb
          0         0         0         0         0       0.433   aux
          0         0         0         0         0       0.495  determiner
          0         0         0         0         0       0.476  invariant
          1         1       0.5         1       0.667       0.498   noun
          0         0         0         0         0       0.494  preposition
          0         0         0         0         0       0.462  pronoun
          0         0         0         0         0        0.05  qualifier
          0         0         0         0         0       0.491   verb
Weighted Avg.   0.5     0.5     0.25     0.5     0.333     0.492
```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
0  0  0  0  0 134  0  0  0  0 |  a = adjective
0  0  0  0  0  43  0  0  0  0 |  b = adverb
0  0  0  0  0  12  0  0  0  0 |  c = aux
0  0  0  0  0 141  0  0  0  0 |  d = determiner
0  0  0  0  0  55  0  0  0  0 |  e = invariant
0  0  0  0  0 642  0  0  0  0 |  f = noun
0  0  0  0  0 162  0  0  0  0 |  g = preposition
0  0  0  0  0  35  0  0  0  0 |  h = pronoun
0  0  0  0  0   1  0  0  0  0 |  i = qualifier
0  0  0  0  0  59  0  0  0  0 |  j = verb

```

Como se puede observar en los resultados, el algoritmo ZeroR queda totalmente descartado para ser el elegido como generador de reglas de desambiguación ya que clasifica todas las palabras como sustantivos (*noun*), lo que explica el bajo porcentaje de acierto.



## 7.1.2 Experimento 2

Para el Experimento 2 se ejecuta la clasificación con el algoritmo **ConjunctiveRule**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Correctly Classified Instances      746          58.0997 %
Incorrectly Classified Instances    538          41.9003 %
Kappa statistic                    0.3301
Mean absolute error                 0.1156
Root mean squared error             0.2407
Relative absolute error             81.7514 %
Root relative squared error         90.6281 %
Total Number of Instances          1284

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0         0         0         0         0         0.644     adjective
      0         0         0         0         0         0.436     adverb
      0         0         0         0         0         0.61      aux
      0         0         0         0         0         0.799     determiner
      0         0         0         0         0         0.684     invariant
      0.93      0.399     0.7         0.93      0.799     0.764     noun
      0.92      0.251     0.346      0.92      0.503     0.824     preposition
      0         0         0         0         0         0.744     pronoun
      0         0         0         0         0         0.27     qualifier
      0         0         0         0         0         0.591     verb
Weighted Avg.   0.581     0.231     0.394     0.581     0.463     0.738

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
0  0  0  0  0 127  7  0  0  0 | a = adjective
0  0  0  0  0  25 18  0  0  0 | b = adverb
0  0  0  0  0  4  8  0  0  0 | c = aux
0  0  0  0  0 15 126  0  0  0 | d = determiner
0  0  0  0  0 15 40  0  0  0 | e = invariant
0  0  0  0  0 597 45  0  0  0 | f = noun
0  0  0  0  0 13 149  0  0  0 | g = preposition
0  0  0  0  0  5 30  0  0  0 | h = pronoun
0  0  0  0  0  1  0  0  0  0 | i = qualifier
0  0  0  0  0 51  8  0  0  0 | j = verb

```

A la luz de los resultados, no se ha mejorado mucho con respecto al algoritmo ZeroR, pues en este caso clasifica las palabras como sustantivos (*noun*) o como preposiciones (*preposition*) y queda también descartado para ser el elegido como generador de reglas de desambiguación. El porcentaje de acierto es aún bastante bajo.

### 7.1.3 Experimento 3

El algoritmo de clasificación utilizado en el Experimento 3 es el llamado **OneR**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.OneR -B 6
Correctly Classified Instances      888                69.1589 %
Incorrectly Classified Instances    396                30.8411 %
Kappa statistic                     0.5026
Mean absolute error                  0.0617
Root mean squared error              0.2484
Relative absolute error              43.6278 %
Root relative squared error          93.5195 %
Total Number of Instances          1284

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.187     0.017     0.568     0.187     0.281     0.585     adjective
      0.279     0.002     0.8       0.279     0.414     0.638     adverb
      0         0.002     0         0         0         0.499     aux
      0.709     0.026     0.769     0.709     0.738     0.841     determiner
      0.527     0.01      0.707     0.527     0.604     0.759     invariant
      0.924     0.478     0.659     0.924     0.769     0.723     noun
      0.685     0.012     0.895     0.685     0.776     0.837     preposition
      0.314     0.002     0.786     0.314     0.449     0.656     pronoun
      0         0         0         0         0         0.5       qualifier
      0.119     0.006     0.5       0.119     0.192     0.556     verb
Weighted Avg.   0.692     0.246     0.688     0.692     0.654     0.723

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
25  0  0  8  3  94  1  0  0  3 | a = adjective
 0 12  0  4  0  24  3  0  0  0 | b = adverb
 0  0  0  0  0  12  0  0  0  0 | c = aux
 5  0  0 100  2  31  3  0  0  0 | d = determiner
 1  0  0  0  29  24  1  0  0  0 | e = invariant
12  2  2  17  5  593  5  3  0  3 | f = noun
 0  0  0  0  0  51 111  0  0  0 | g = preposition
 0  1  0  0  0  22  0 11  0  1 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 1  0  0  1  2  48  0  0  0  7 | j = verb

```

Observando los resultados obtenidos con el algoritmo OneR, se puede comprobar una importante mejoría en los aciertos ya que ahora las palabras se clasifican en todas las clases, excepto en cualificadores (*qualifier*). Se puede conseguir mayor porcentaje de instancias correctas, por lo que continúa la búsqueda de un algoritmo de mayor eficacia.

## 7.1.4 Experimento 4

En el Experimento 4 se efectúa la clasificación con el algoritmo **DecisionTable**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Correctly Classified Instances      903      70.3271 %
Incorrectly Classified Instances    381      29.6729 %
Kappa statistic                    0.531
Mean absolute error                 0.1029
Root mean squared error             0.2153
Relative absolute error             72.7641 %
Root relative squared error         81.078 %
Total Number of Instances          1284
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.082    0.012    0.44      0.082    0.138      0.726  adjective
      0.163     0      1        0.163    0.28      0.751  adverb
      0.083    0.001    0.5      0.083    0.143      0.907  aux
      0.787    0.064    0.603    0.787    0.683      0.938  determiner
      0.527    0.005    0.829    0.527    0.644      0.86   invariant
      0.931    0.399    0.7      0.931    0.799      0.795  noun
      0.778    0.024    0.824    0.778    0.8        0.959  preposition
      0.4      0.002    0.824    0.4      0.538      0.839  pronoun
      0        0        0        0        0          0.027  qualifier
      0.102    0.001    0.857    0.102    0.182      0.67   verb
Weighted Avg.  0.703    0.211    0.702    0.703    0.651      0.821
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
11  0  0  6  0 117  0  0  0  0 | a = adjective
 0  7  0 12  2  19  2  1  0  0 | b = adverb
 0  0  1  0  0  10  1  0  0  0 | c = aux
 0  0  0 111  2  18  9  1  0  0 | d = determiner
 0  0  0  4  29  18  4  0  0  0 | e = invariant
12  0  1  19  1 598  9  1  0  1 | f = noun
 0  0  0 22  0  14 126  0  0  0 | g = preposition
 0  0  0 10  1  8  2  14  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 2  0  0  0  0  51  0  0  0  6 | j = verb
```

Con el algoritmo DecisionTable se ha conseguido una leve mejoría con respecto a OneR, pero aún puede incrementarse el porcentaje de aciertos, por lo que se procede a la ejecución de un nuevo experimento con otro algoritmo diferente.

## 7.1.5 Experimento 5

Para el Experimento 5 se ejecuta la clasificación con el algoritmo **Ridor**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.Ridor -F 3 -S 1 -N 2.0
Correctly Classified Instances      910      70.8723 %
Incorrectly Classified Instances    374      29.1277 %
Kappa statistic                    0.5631
Mean absolute error                 0.0583
Root mean squared error             0.2414
Relative absolute error             41.204 %
Root relative squared error        90.8847 %
Total Number of Instances         1284

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.269     0.054     0.367     0.269     0.31        0.607     adjective
      0.395     0.015     0.486     0.395     0.436       0.69      adverb
      0.5        0.004     0.545     0.5        0.522       0.748     aux
      0.78       0.011     0.894     0.78       0.833       0.884     determiner
      0.545      0.006     0.811     0.545     0.652       0.77      invariant
      0.846      0.33      0.719     0.846     0.777       0.758     noun
      0.877      0.028     0.821     0.877     0.848       0.924     preposition
      0.4        0.004     0.737     0.4        0.519       0.698     pronoun
      0          0         0         0         0           0.5       qualifier
      0.203      0.017     0.364     0.203     0.261       0.593     verb
Weighted Avg.   0.709     0.177     0.693     0.709     0.693       0.766

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
36  0  1  0  2  87  6  0  0  2 | a = adjective
 2 17  0  2  0 17  3  2  0  0 | b = adverb
 1  0  6  0  1  4  0  0  0  0 | c = aux
 3  1  0 110  1 17  8  0  0  1 | d = determiner
 3  0  0  3 30 18  1  0  0  0 | e = invariant
49 11  4  4  1 543 10  3  0 17 | f = noun
 0  1  0  0  1 17 142  0  0  1 | g = preposition
 0  3  0  4  1 10  3 14  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 4  2  0  0  0 41  0  0  0 12 | j = verb

```

Se ha incrementado en casi un 0,55% el porcentaje de acierto con respecto al anterior algoritmo. Aún así, se van a seguir probando algoritmos de clasificación en pos de un mejor resultado.

## 7.1.6 Experimento 6

El algoritmo de clasificación utilizado en el Experimento 6 es el **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Correctly Classified Instances      960                74.7664 %
Incorrectly Classified Instances    324                25.2336 %
Kappa statistic                    0.6174
Mean absolute error                 0.066
Root mean squared error             0.2042
Relative absolute error             46.6953 %
Root relative squared error         76.8734 %
Total Number of Instances          1284

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.224    0.03    0.462    0.224    0.302    0.779    adjective
      0.372    0.006    0.667    0.372    0.478    0.724    adverb
      0.5       0.002    0.75     0.5     0.6     0.828    aux
      0.844    0.018    0.85     0.844    0.847    0.953    determiner
      0.636    0.006    0.833    0.636    0.722    0.827    invariant
      0.902    0.307    0.746    0.902    0.817    0.826    noun
      0.914    0.026    0.836    0.914    0.873    0.97     preposition
      0.514    0.006    0.692    0.514    0.59     0.782    pronoun
      0        0        0        0        0        0.496    qualifier
      0.153    0.014    0.346    0.153    0.212    0.706    verb
Weighted Avg.  0.748    0.163    0.72     0.748    0.721    0.843

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
30  0  0  0  0  98  3  0  0  3 | a = adjective
 1 16  0  2  0 17  5  1  0  1 | b = adverb
 0  0  6  2  1  2  0  1  0  0 | c = aux
 0  0  2 119  0  8  8  1  0  3 | d = determiner
 0  1  0  3 35 13  3  0  0  0 | e = invariant
27  3  0  5  5 579  8  5  0 10 | f = noun
 0  2  0  4  0  8 148  0  0  0 | g = preposition
 1  2  0  5  1  6  2 18  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 6  0  0  0  0 44  0  0  0  9 | j = verb

```

Se ha visto incrementado en casi un 4% el porcentaje de acierto con respecto al anterior algoritmo (Ridor). Queda por probar el algoritmo J48 y algunos meta-algoritmos, que consiguen superar los resultados obtenidos hasta el momento.

## 7.1.7 Experimento 7

En el Experimento 7 se efectúa la clasificación con el meta-algoritmo **Bagging** basado en el algoritmo de árboles **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances          968                75.3894 %
Incorrectly Classified Instances        316                24.6106 %
Kappa statistic                        0.6256
Mean absolute error                    0.0656
Root mean squared error                0.1931
Relative absolute error                46.403 %
Root relative squared error            72.6965 %
Total Number of Instances              1284
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.254    0.032    0.479    0.254    0.332    0.837    adjective
      0.419    0.006    0.72     0.419    0.529    0.799    adverb
      0.5      0.002    0.667    0.5      0.571    0.904    aux
      0.858    0.016    0.871    0.858    0.864    0.966    determiner
      0.636    0.007    0.795    0.636    0.707    0.883    invariant
      0.905    0.315    0.742    0.905    0.815    0.852    noun
      0.914    0.019    0.876    0.914    0.894    0.979    preposition
      0.486    0.004    0.773    0.486    0.596    0.855    pronoun
      0        0        0        0        0        0.458    qualifier
      0.136    0.011    0.364    0.136    0.198    0.727    verb
Weighted Avg. 0.754    0.166    0.729    0.754    0.729    0.873
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
34  0  0  0  0  95  3  0  0  2 | a = adjective
 0 18  0  1  1 19  2  2  0  0 | b = adverb
 0  0  6  0  1  4  1  0  0  0 | c = aux
 1  0  2 121  1  8  6  1  0  1 | d = determiner
 0  1  0  2 35 15  2  0  0  0 | e = invariant
32  3  1  5  2 581  6  2  0 10 | f = noun
 0  0  0  3  3  7 148  0  0  1 | g = preposition
 1  3  0  7  1  5  1 17  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 3  0  0  0  0 48  0  0  0  8 | j = verb
```

Esta vez, con este meta-algoritmo de bagging, se ha conseguido una mejora de más de 0,6% con respecto al Part, aunque resulta un tanto decepcionante que sólo haya conseguido llegar a ese valor.

## 7.1.8 Experimento 8

Para el Experimento 8 se ejecuta la clasificación con el meta-algoritmo **MultiBoostAB** apoyado en el algoritmo de árboles **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.MultiBoostAB -C 3 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      973                75.7788 %
Incorrectly Classified Instances    311                24.2212 %
Kappa statistic                    0.6378
Mean absolute error                 0.0487
Root mean squared error             0.2108
Relative absolute error             34.4554 %
Root relative squared error         79.3623 %
Total Number of Instances          1284

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.358    0.041    0.505    0.358    0.419    0.799    adjective
      0.442    0.007    0.679    0.442    0.535    0.802    adverb
      0.5      0.002    0.75     0.5      0.6      0.798    aux
      0.858    0.018    0.852    0.858    0.855    0.972    determiner
      0.636    0.007    0.795    0.636    0.707    0.862    invariant
      0.882    0.283    0.757    0.882    0.814    0.861    noun
      0.92     0.016    0.892    0.92     0.906    0.986    preposition
      0.571    0.007    0.69     0.571    0.625    0.865    pronoun
      0        0        0        0        0        0.703    qualifier
      0.153    0.011    0.391    0.153    0.22     0.647    verb
Weighted Avg. 0.758    0.151    0.738    0.758    0.74     0.87

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a   b   c   d   e   f   g   h   i   j   <-- classified as
48  0   0   0   1  82   2   0   0   1 | a = adjective
 1 19   0   2   0 16   2   2   0   1 | b = adverb
 0   0   6   2   1   2   1   0   0   0 | c = aux
 0   0   1 121   0 11   6   2   0   0 | d = determiner
 1   1   0   3  35 11   2   2   0   0 | e = invariant
43  3   1   5   6 566   4   3   0 11 | f = noun
 0   2   0   4   0   6 149   0   0   1 | g = preposition
 0   3   0   5   1   5   1  20   0   0 | h = pronoun
 0   0   0   0   0   1   0   0   0   0 | i = qualifier
 2   0   0   0   0  48   0   0   0   9 | j = verb

```

El porcentaje de acierto utilizando este meta-algoritmo de boosting se ha incrementado casi más de un 0,4% con respecto al previo meta-algoritmo de bagging sobre J48, pero aún así se continúa obteniendo valores muy bajos.

## 7.1.9 Experimento 9

El algoritmo de clasificación utilizado en el Experimento 9 es el **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Correctly Classified Instances      976                76.0125 %
Incorrectly Classified Instances    308                23.9875 %
Kappa statistic                    0.6306
Mean absolute error                 0.0686
Root mean squared error             0.2
Relative absolute error             48.5552 %
Root relative squared error         75.2958 %
Total Number of Instances          1284
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.224     0.027     0.492     0.224     0.308       0.728     adjective
      0.372     0.006     0.696     0.372     0.485       0.764     adverb
      0.5        0.001     0.857     0.5        0.632       0.818     aux
      0.858     0.018     0.852     0.858     0.855       0.951     determiner
      0.618     0.005     0.85      0.618     0.716       0.845     invariant
      0.927     0.329     0.738     0.927     0.822       0.819     noun
      0.926     0.019     0.877     0.926     0.901       0.969     preposition
      0.486     0.005     0.739     0.486     0.586       0.748     pronoun
      0         0         0         0         0           0.359     qualifier
      0.119     0.003     0.636     0.119     0.2         0.67      verb
Weighted Avg.   0.76      0.172     0.742     0.76      0.729       0.833
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
30  0  0  1  1 99  2  0  0  1 | a = adjective
 1 16  0  3  0 17  3  2  0  1 | b = adverb
 0  0  6  1  1  3  1  0  0  0 | c = aux
 0  0  1 121  0 12  6  1  0  0 | d = determiner
 0  1  0  3 34 15  2  0  0  0 | e = invariant
27  2  0  4  3 595  6  3  0  2 | f = noun
 0  0  0  4  0  8 150  0  0  0 | g = preposition
 1  4  0  5  1  6  1 17  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 2  0  0  0  0 50  0  0  0  7 | j = verb
```

Esta vez, con este algoritmo de árboles, se ha conseguido una mejora de, al menos, 0,25% con respecto a meta-algoritmos basados en J48. Queda por realizar experimentos con otro algoritmo de boosting (AdaBoostM1) y el Bagging tomando como referencia el algoritmo Part.



## 7.1.10 Experimento 10

En el Experimento 10 se efectúa la clasificación con el meta-algoritmo **Bagging**, tomando como base el algoritmo **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      983                76.5576 %
Incorrectly Classified Instances    301                23.4424 %
Kappa statistic                    0.6444
Mean absolute error                 0.0651
Root mean squared error             0.1921
Relative absolute error              46.066 %
Root relative squared error          72.3209 %
Total Number of Instances          1284
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.276    0.029    0.529    0.276    0.363    0.846    adjective
      0.395    0.005    0.739    0.395    0.515    0.767    adverb
      0.5      0.002    0.75     0.5      0.6      0.817    aux
      0.872    0.016    0.872    0.872    0.872    0.958    determiner
      0.655    0.004    0.878    0.655    0.75     0.869    invariant
      0.908    0.302    0.75     0.908    0.822    0.856    noun
      0.938    0.02     0.869    0.938    0.902    0.978    preposition
      0.571    0.006    0.714    0.571    0.635    0.871    pronoun
      0        0.001    0        0        0        0.484    qualifier
      0.153    0.009    0.45     0.153    0.228    0.726    verb
Weighted Avg. 0.766    0.159    0.745    0.766    0.741    0.873
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
37  0  0  0  0  94  3  0  0  0 | a = adjective
 0 17  0  3  0 18  2  2  0  1 | b = adverb
 0  0  6  2  0  3  1  0  0  0 | c = aux
 0  0  1 123  1  7  7  1  0  1 | d = determiner
 0  1  0  2 36 15  1  0  0  0 | e = invariant
30  1  1  4  3 583  7  4  1  8 | f = noun
 0  1  0  2  0  5 152  1  0  1 | g = preposition
 0  3  0  5  1  4  2 20  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 3  0  0  0  0  47  0  0  0  9 | j = verb
```

La aplicación del meta-algoritmo de bagging al algoritmo Part supone una mejora de más de 1,75% sobre el Part normal. Como se está viendo en estos últimos experimentos, las mejoras son relativamente pobres, el algoritmo elegido esta cercano.

## 7.1.11 Experimento 11

Para el Experimento 11 se ejecuta la clasificación con el meta-algoritmo **AdaBoostM1** basado en el **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances          991                77.1807 %
Incorrectly Classified Instances        293                22.8193 %
Kappa statistic                        0.6702
Mean absolute error                    0.048
Root mean squared error                0.2024
Relative absolute error                 33.9284 %
Root relative squared error            76.2059 %
Total Number of Instances              1284
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.485    0.049    0.537    0.485    0.51       0.846    adjective
      0.512    0.01     0.629    0.512    0.564    0.776    adverb
      0.583    0.002    0.778    0.583    0.667    0.869    aux
      0.901    0.015    0.882    0.901    0.891    0.967    determiner
      0.673    0.011    0.74     0.673    0.705    0.87     invariant
      0.844    0.213    0.798    0.844    0.821    0.87     noun
      0.944    0.013    0.911    0.944    0.927    0.984    preposition
      0.571    0.009    0.645    0.571    0.606    0.862    pronoun
      0       0.001    0       0       0       0.528    qualifier
      0.305    0.023    0.391    0.305    0.343    0.73     verb
Weighted Avg. 0.772    0.117    0.763    0.772    0.766    0.882
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
65  1  0  0  1  61  1  0  0  5 | a = adjective
 1 22  0  2  0 13  3  1  0  1 | b = adverb
 0  0  7  1  1  2  1  0  0  0 | c = aux
 0  0  1 127  1  6  2  3  0  1 | d = determiner
 1  1  0  3 37  9  2  2  0  0 | e = invariant
50  7  1  5  7 542  4  5  1 20 | f = noun
 0  1  0  2  2  3 153  0  0  1 | g = preposition
 1  3  0  4  1  5  1 20  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 3  0  0  0  0 37  1  0  0 18 | j = verb
```

El porcentaje de acierto utilizando este meta-algoritmo de boosting sobre J48 se ha incrementado en algo más de un 1% con respecto al J48 normal. Este es un claro candidato a ser el algoritmo elegido para obtener las reglas de desambiguación.

## 7.1.12 Experimento 12

El meta-algoritmo de clasificación utilizado en el Experimento 12 es el **AdaBoostM1**, apoyado en el algoritmo **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances          992                77.2586 %
Incorrectly Classified Instances        292                22.7414 %
Kappa statistic                        0.6698
Mean absolute error                    0.0485
Root mean squared error                0.2021
Relative absolute error                34.2898 %
Root relative squared error            76.0891 %
Total Number of Instances              1284
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.463    0.043    0.554    0.463    0.504    0.825    adjective
      0.512    0.014    0.564    0.512    0.537    0.746    adverb
      0.5      0.002    0.75     0.5      0.6      0.863    aux
      0.901    0.017    0.87     0.901    0.885    0.968    determiner
      0.673    0.011    0.74     0.673    0.705    0.885    invariant
      0.854    0.22     0.795    0.854    0.823    0.858    noun
      0.944    0.013    0.911    0.944    0.927    0.987    preposition
      0.571    0.006    0.714    0.571    0.635    0.827    pronoun
      0        0.002    0        0        0        0.587    qualifier
      0.288    0.02     0.405    0.288    0.337    0.678    verb
Weighted Avg. 0.773    0.12     0.762    0.773    0.765    0.871
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
62  2  0  0  1  64  1  0  0  4 | a = adjective
 1 22  0  2  0 13  3  1  0  1 | b = adverb
 0  0  6  2  1  2  1  0  0  0 | c = aux
 0  0  1 127  1  6  2  3  0  1 | d = determiner
 2  1  0  2 37  9  2  2  0  0 | e = invariant
45  9  1  6  7 548  4  2  2 18 | f = noun
 0  1  0  2  2  3 153  0  0  1 | g = preposition
 0  3  0  5  1  4  2 20  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 2  1  0  0  0 39  0  0  0 17 | j = verb
```

Con este meta-algoritmo de boosting, se ha conseguido una mejora de casi 2,5% con respecto al Part básico. Tanto este como el mismo meta-algoritmo basado en el J48 son los dos máximos candidatos a ser elegidos para crear las reglas de desambiguación.

### 7.1.13 Conclusiones de la Primera Fase de Experimentación

A la vista de los resultados obtenidos con todos los algoritmos, se decide escoger el meta-algoritmo AdaBoostM1 ejecutado sobre la clasificación Part, dado que es el que mejor resultado obtiene y es un modelo basado en reglas, mientras que el AdaBoostM1 ejecutado sobre J48 el resultado viene expresado en forma de árbol de decisión. Con las reglas de desambiguación generadas, se espera mejorar los resultados en una segunda fase, donde también se tendrá en cuenta el contexto. Aún así, es necesario mencionar el bajo éxito de los experimentos realizados.

A continuación se muestra una gráfica comparando los porcentajes de acierto de los algoritmos utilizados en la Primera Fase de Experimentación:

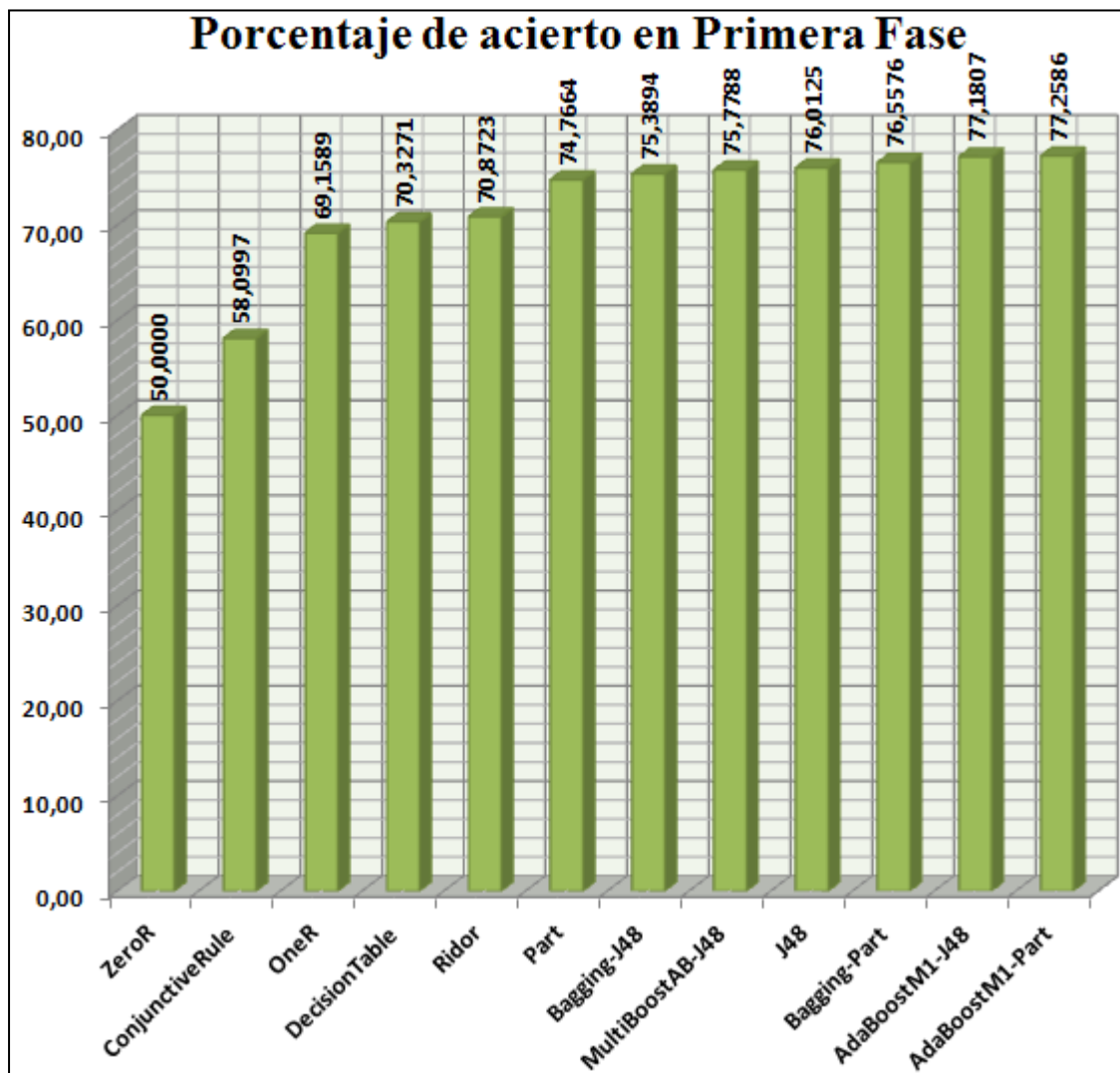


Figura 28: Gráfica de comparación de los algoritmos en la 1ª fase de experimentación

## 7.2 Segunda Fase de Experimentación

### 7.2.1 Experimentos con número de regla sólo para la palabra (F2-1R)

#### 7.2.1.1 Experimento 1

En el Experimento 1 se efectúa la clasificación con el algoritmo **ZeroR**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.ZeroR
Correctly Classified Instances      641          50      %
Incorrectly Classified Instances    641          50      %
Kappa statistic                    0
Mean absolute error                 0.1414
Root mean squared error             0.2656
Relative absolute error             100      %
Root relative squared error         100      %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0         0         0           0         0           0.489     adjective
      0         0         0           0         0           0.474     adverb
      0         0         0           0         0           0.433     aux
      0         0         0           0         0           0.496     determiner
      0         0         0           0         0           0.476     invariant
      1         1         0.5         1         0.667       0.499     noun
      0         0         0           0         0           0.497     preposition
      0         0         0           0         0           0.463     pronoun
      0         0         0           0         0           0.05      qualifier
      0         0         0           0         0           0.492     verb
Weighted Avg.   0.5       0.5       0.25      0.5       0.333     0.493
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
0  0  0  0  0 134  0  0  0  0 | a = adjective
0  0  0  0  0  43  0  0  0  0 | b = adverb
0  0  0  0  0  12  0  0  0  0 | c = aux
0  0  0  0  0 141  0  0  0  0 | d = determiner
0  0  0  0  0  55  0  0  0  0 | e = invariant
0  0  0  0  0 641  0  0  0  0 | f = noun
0  0  0  0  0 161  0  0  0  0 | g = preposition
0  0  0  0  0  35  0  0  0  0 | h = pronoun
0  0  0  0  0  1  0  0  0  0 | i = qualifier
0  0  0  0  0  59  0  0  0  0 | j = verb
```

El resultado no mejora el porcentaje obtenido por el mismo algoritmo en la primera fase, sino que lo iguala prácticamente.

### 7.2.1.2 Experimento 2

Para el Experimento 2 se ejecuta la clasificación con el algoritmo **ConjunctiveRule**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Correctly Classified Instances      767          59.8284 %
Incorrectly Classified Instances    515          40.1716 %
Kappa statistic                    0.2691
Mean absolute error                 0.1202
Root mean squared error             0.2453
Relative absolute error             85.0017 %
Root relative squared error         92.3472 %
Total Number of Instances          1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0        0        0          0        0          0.554   adjective
      0        0        0          0        0          0.513   adverb
      0        0        0          0        0          0.422   aux
      0.929    0.028    0.804    0.929    0.862    0.943   determiner
      0        0        0          0        0          0.525   invariant
      0.992    0.754    0.568    0.992    0.723    0.62    noun
      0        0        0          0        0          0.505   preposition
      0        0        0          0        0          0.479   pronoun
      0        0        0          0        0          0.107   qualifier
      0        0        0          0        0          0.561   verb
Weighted Avg.  0.598    0.38    0.373    0.598    0.456    0.618

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
0  0  0  1  0 133  0  0  0  0 | a = adjective
0  0  0  2  0  41  0  0  0  0 | b = adverb
0  0  0  2  0  10  0  0  0  0 | c = aux
0  0  0 131  0  10  0  0  0  0 | d = determiner
0  0  0  2  0  53  0  0  0  0 | e = invariant
0  0  0  5  0 636  0  0  0  0 | f = noun
0  0  0 17  0 144  0  0  0  0 | g = preposition
0  0  0  3  0  32  0  0  0  0 | h = pronoun
0  0  0  0  0  1  0  0  0  0 | i = qualifier
0  0  0  0  0  59  0  0  0  0 | j = verb

```

En esta segunda fase sí se han incrementado el porcentaje de aciertos en más de un 1,7% con este algoritmo con respecto a los obtenidos con el mismo en la primera fase.

### 7.2.1.3 Experimento 3

El algoritmo de clasificación utilizado en el Experimento 3 es el llamado **OneR**.

Los resultados de la clasificación son:

```
Scheme: weka.classifiers.rules.OneR -B 6
Correctly Classified Instances      891                69.5008 %
Incorrectly Classified Instances    391                30.4992 %
Kappa statistic                    0.5248
Mean absolute error                 0.061
Root mean squared error             0.247
Relative absolute error             43.1406 %
Root relative squared error         92.9961 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.157    0.027    0.404    0.157    0.226    0.565    adjective
      0.209    0.005    0.6      0.209    0.31     0.602    adverb
      0.083    0.005    0.143    0.083    0.105    0.539    aux
      0.801    0.042    0.702    0.801    0.748    0.88     determiner
      0.545    0.007    0.769    0.545    0.638    0.769    invariant
      0.906    0.385    0.702    0.906    0.791    0.761    noun
      0.702    0.032    0.758    0.702    0.729    0.835    preposition
      0.429    0      1      0.429    0.6      0.714    pronoun
      0      0      0      0      0      0.5      qualifier
      0.136    0.007    0.5      0.136    0.213    0.565    verb
Weighted Avg. 0.695    0.205    0.67     0.695    0.658    0.745
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
21  0  0  3  0 109  1  0  0  0 | a = adjective
 1  9  0  2  1  24  6  0  0  0 | b = adverb
 1  0  1  0  1  9   0  0  0  0 | c = aux
 1  0  0 113  0  17  10  0  0  0 | d = determiner
 1  0  0  2  30  19  3  0  0  0 | e = invariant
20  2  4  14  2 581  12  0  0  6 | f = noun
 2  3  1  20  5  17 113  0  0  0 | g = preposition
 1  1  0  6  0  8   2  15  0  2 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 4  0  1  1  0  43  2  0  0  8 | j = verb
```

Los resultados son casi un 0.35% mejor que con el mismo algoritmo en la primera fase, por lo que esta vez se ha conseguido optimizar la clasificación, aunque no sea mucha la diferencia.

### 7.2.1.4 Experimento 4

En el Experimento 4 se efectúa la clasificación con el algoritmo **Ridor**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.Ridor -F 3 -S 1 -N 2.0
Correctly Classified Instances      936                73.0109 %
Incorrectly Classified Instances    346                26.9891 %
Kappa statistic                    0.6027
Mean absolute error                 0.054
Root mean squared error             0.2323
Relative absolute error             38.1756 %
Root relative squared error         87.4811 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.276    0.027    0.544    0.276    0.366    0.625    adjective
      0.465    0.019    0.455    0.465    0.46    0.723    adverb
      0.583    0.006    0.467    0.583    0.519    0.789    aux
      0.801    0.034    0.743    0.801    0.771    0.884    determiner
      0.691    0.015    0.679    0.691    0.685    0.838    invariant
      0.872    0.256    0.773    0.872    0.82    0.808    noun
      0.764    0.023    0.826    0.764    0.794    0.87    preposition
      0.457    0.008    0.615    0.457    0.525    0.725    pronoun
      0        0        0        0        0        0.5    qualifier
      0.39     0.021    0.469    0.39    0.426    0.684    verb
Weighted Avg.  0.73     0.14    0.716    0.73    0.716    0.795
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
37  1  0  3  0  87  2  0  0  4 | a = adjective
 1 20  0  4  5 10  3  0  0  0 | b = adverb
 1  0  7  1  2  1  0  0  0  0 | c = aux
 1  2  0 113  1 11  6  6  0  1 | d = determiner
 0  1  0  3 38  9  3  0  0  1 | e = invariant
24 14  7  4  5 559  8  1  0 19 | f = noun
 3  4  0 17  4  7 123  3  0  0 | g = preposition
 0  2  0  7  1  4  4 16  0  1 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 1  0  1  0  0 34  0  0  0 23 | j = verb
```

Esta vez, el algoritmo Ridor ha aumentado su éxito en casi un 2,1% con respecto al resultado de la primera fase.



### 7.2.1.5 Experimento 5

Para el Experimento 5 se ejecuta la clasificación con el algoritmo **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Correctly Classified Instances      942          73.4789 %
Incorrectly Classified Instances    340          26.5211 %
Kappa statistic                    0.6092
Mean absolute error                 0.0711
Root mean squared error             0.2056
Relative absolute error             50.2974 %
Root relative squared error         77.4133 %
Total Number of Instances          1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.388    0.045    0.5         0.388   0.437     0.791   adjective
      0.326    0.011    0.5         0.326   0.394     0.764   adverb
      0         0.002    0         0         0         0.551   aux
      0.901    0.032    0.779      0.901   0.836     0.962   determiner
      0.727    0.017    0.656      0.727   0.69      0.927   invariant
      0.86     0.265    0.764      0.86    0.809     0.837   noun
      0.696    0.018    0.848      0.696   0.765     0.907   preposition
      0.657    0.01     0.639      0.657   0.648     0.854   pronoun
      0         0         0         0         0         0.498   qualifier
      0.39     0.009    0.676      0.39    0.495     0.802   verb
Weighted Avg. 0.735    0.145    0.72       0.735   0.721     0.852

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
52  0  0  2  0  80  0  0  0  0 | a = adjective
 1 14  0  3  1  21  1  1  0  1 | b = adverb
 1  0  0  2  0  6  3  0  0  0 | c = aux
 0  0  0 127  0  6  4  4  0  0 | d = determiner
 0  1  0  0  40 10  4  0  0  0 | e = invariant
41  6  2  8 12 551  7  4  0 10 | f = noun
 5  2  1 18  7 12 112  4  0  0 | g = preposition
 1  3  0  2  1  4  1 23  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 3  2  0  1  0 30  0  0  0 23 | j = verb

```

Se ha experimentado un empeoramiento con respecto al resultado de Part en la primera fase de casi un 1,3%, por lo que se puede decir que no ha sido un éxito la desambiguación en este caso.

### 7.2.1.6 Experimento 6

El algoritmo de clasificación utilizado en el Experimento 6 es el **DecisionTable**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Correctly Classified Instances          944                73.6349 %
Incorrectly Classified Instances        338                26.3651 %
Kappa statistic                        0.5871
Mean absolute error                    0.0917
Root mean squared error                0.201
Relative absolute error                64.8427 %
Root relative squared error            75.6972 %
Total Number of Instances             1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.09     0.01     0.5       0.09    0.152     0.823   adjective
      0.326    0.013    0.467    0.326    0.384     0.86    adverb
      0.083    0.004    0.167    0.083    0.111     0.984    aux
      0.943    0.032    0.787    0.943    0.858     0.976   determiner
      0.636    0.016    0.636    0.636    0.636     0.923   invariant
      0.95     0.367    0.722    0.95     0.82     0.838    noun
      0.72     0.01     0.913    0.72     0.806     0.96   preposition
      0.486    0.002    0.85     0.486    0.618     0.964   pronoun
      0        0        0        0        0        0.957   qualifier
      0.119    0        1        0.119    0.212     0.82    verb
Weighted Avg.  0.736    0.19    0.728    0.736    0.689     0.875
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
12  0  0  1  0 120  1  0  0  0 |  a = adjective
 0 14  0  2  1  24  2  0  0  0 |  b = adverb
 2  0  1  2  0  6  1  0  0  0 |  c = aux
 2  0  0 133  1  3  0  2  0  0 |  d = determiner
 0  1  0  4 35 13  2  0  0  0 |  e = invariant
 0  9  4  5 13 609  0  1  0  0 |  f = noun
 5  5  0 17  4 14 116  0  0  0 |  g = preposition
 2  0  0  5  1  5  5 17  0  0 |  h = pronoun
 0  0  0  0  0  1  0  0  0  0 |  i = qualifier
 1  1  1  0  0 49  0  0  0  7 |  j = verb
```

Se ha incrementado en poco más de un 3,3% el resultado de aciertos con respecto al resultado del algoritmo de la primera fase, una mejora reseñable.

### 7.2.1.7 Experimento 7

En el Experimento 7 se efectúa la clasificación con el algoritmo de árboles **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Correctly Classified Instances      944                73.6349 %
Incorrectly Classified Instances    338                26.3651 %
Kappa statistic                    0.5934
Mean absolute error                 0.0771
Root mean squared error             0.2033
Relative absolute error             54.5547 %
Root relative squared error         76.561 %
Total Number of Instances          1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.216    0.021    0.547    0.216    0.31       0.771    adjective
      0.186    0.003    0.667    0.186    0.291      0.688    adverb
      0        0.001    0        0        0         0.57     aux
      0.915    0.03     0.791    0.915    0.849      0.954    determiner
      0.673    0.01     0.755    0.673    0.712      0.832    invariant
      0.922    0.348    0.726    0.922    0.812      0.812    noun
      0.671    0.014    0.871    0.671    0.758      0.906    preposition
      0.8       0.014    0.622    0.8      0.7        0.898    pronoun
      0        0        0        0        0         0.498    qualifier
      0.237    0.006    0.667    0.237    0.35       0.841    verb
Weighted Avg. 0.736    0.182    0.719    0.736    0.703      0.833

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
29  0  0  1  0 102  1  0  0  1 | a = adjective
 0  8  0  2  1  29  0  2  0  1 | b = adverb
 0  0  0  2  0  7   3  0  0  0 | c = aux
 0  0  0 129  0  5   3  4  0  0 | d = determiner
 0  1  0  2  37 11   4  0  0  0 | e = invariant
20  3  0  7  7 591  4  4  0  5 | f = noun
 4  0  1 17  4 20 108  7  0  0 | g = preposition
 0  0  0  3  0  3   1 28  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 0  0  0  0  0 45  0  0  0 14 | j = verb

```

Con el algoritmo J48 se ha igualado el resultado obtenido con el algoritmo DecisionTable en esta misma fase, pero ha supuesto una caída de efectividad de casi 2,15% con respecto a la clasificación del J48 en la primera fase, es un gran paso atrás.

### 7.2.1.8 Experimento 8

Para el Experimento 8 se ejecuta la clasificación con el meta-algoritmo **AdaBoostM1** basado en el **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances          955                74.493 %
Incorrectly Classified Instances        327                25.507 %
Kappa statistic                        0.626
Mean absolute error                    0.0675
Root mean squared error                0.1972
Relative absolute error                 47.7148 %
Root relative squared error            74.2464 %
Total Number of Instances              1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.351    0.041     0.5        0.351    0.412     0.823   adjective
      0.395    0.006     0.68       0.395     0.5      0.832   adverb
      0.167    0.005     0.25       0.167     0.2      0.772   aux
      0.844    0.025     0.81       0.844     0.826    0.971  determiner
      0.691    0.012     0.717     0.691     0.704    0.946  invariant
      0.861    0.246     0.777     0.861     0.817     0.88   noun
      0.776    0.033     0.772     0.776     0.774     0.92  preposition
      0.686    0.009     0.686     0.686     0.686    0.946  pronoun
      0        0         0         0         0      0.468  qualifier
      0.525    0.014     0.646     0.525     0.579    0.856  verb
Weighted Avg.  0.745    0.136     0.731     0.745     0.734     0.89

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
47  1  0  3  0 80  1  0  0  2 |  a = adjective
 2 17  0  2  1 16  3  1  0  1 |  b = adverb
 0  0  2  1  0  7  2  0  0  0 |  c = aux
 1  0  0 119  0  7  9  5  0  0 |  d = determiner
 0  1  0  1 38  8  6  1  0  0 |  e = invariant
40  4  3  5 10 552 12  1  0 14 |  f = noun
 2  1  2 13  3 12 125  3  0  0 |  g = preposition
 0  1  0  3  1  2  4 24  0  0 |  h = pronoun
 0  0  0  0  0  1  0  0  0  0 |  i = qualifier
 2  0  1  0  0 25  0  0  0 31 |  j = verb

```

Con este meta-algoritmo se ha bajado un 2.25% de efectividad con respecto al resultado obtenido en la primera fase, que además fue el que mayor valor consiguió.

### 7.2.1.9 Experimento 9

El meta-algoritmo de clasificación utilizado en el Experimento 9 es el **Bagging**, apoyado en el algoritmo **J48**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances          960                74.883 %
Incorrectly Classified Instances        322                25.117 %
Kappa statistic                        0.6148
Mean absolute error                     0.0747
Root mean squared error                 0.1962
Relative absolute error                 52.8606 %
Root relative squared error             73.894 %
Total Number of Instances              1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.231    0.02    0.574    0.231    0.33      0.847  adjective
      0.186    0.002    0.727    0.186    0.296    0.777  adverb
      0        0.001    0        0        0        0.857  aux
      0.922    0.03    0.793    0.922    0.852    0.971  determiner
      0.691    0.01    0.76    0.691    0.724    0.922  invariant
      0.924    0.329    0.737    0.924    0.82     0.85   noun
      0.696    0.012    0.889    0.696    0.78     0.934  preposition
      0.771    0.013    0.628    0.771    0.692    0.901  pronoun
      0        0        0        0        0        0.498  qualifier
      0.373    0.007    0.733    0.373    0.494    0.876  verb
Weighted Avg.  0.749    0.173    0.735    0.749    0.719    0.877

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
31  0  0  1  0 101  1  0  0  0 |  a = adjective
 0  8  0  2  1 29  0  2  0  1 |  b = adverb
 0  0  0  2  0  7  3  0  0  0 |  c = aux
 0  0  0 130  0  6  1  4  0  0 |  d = determiner
 0  1  0  2 38 10  3  1  0  0 |  e = invariant
20  2  0  6  6 592  4  4  0  7 |  f = noun
 3  0  1 17  4 19 112  5  0  0 |  g = preposition
 0  0  0  4  1  2  1 27  0  0 |  h = pronoun
 0  0  0  0  0  1  0  0  0  0 |  i = qualifier
 0  0  0  0  0 36  1  0  0 22 |  j = verb

```

Continúa empeorando el porcentaje de éxito, esta vez con un 0,5% de decremento comparado con el resultado del mismo meta-algoritmo en la primera fase.

### 7.2.1.10 Experimento 10

En el Experimento 10 se efectúa la clasificación con el meta-algoritmo **AdaBoostM1**, tomando como base el algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme: weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      963                75.117 %
Incorrectly Classified Instances    319                24.883 %
Kappa statistic                    0.6328
Mean absolute error                 0.0663
Root mean squared error             0.1959
Relative absolute error              46.8829 %
Root relative squared error         73.7502 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.343    0.044    0.474    0.343    0.398     0.833  adjective
      0.442    0.008    0.655    0.442    0.528     0.836  adverb
      0         0.002    0         0         0         0.86   aux
      0.865    0.022    0.83     0.865    0.847     0.967  determiner
      0.709    0.007    0.83     0.709    0.765     0.956  invariant
      0.878    0.246    0.781    0.878    0.827     0.882  noun
      0.776    0.034    0.767    0.776    0.772     0.903  preposition
      0.629    0.007    0.71     0.629    0.667     0.919  pronoun
      0         0         0         0         0         0.077  qualifier
      0.458    0.015    0.6      0.458    0.519     0.84   verb
Weighted Avg.  0.751    0.136    0.732    0.751    0.738     0.889
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
46  1  0  1  0 82  2  0  0  2 | a = adjective
 3 19  0  2  2 12  3  1  0  1 | b = adverb
 0  0  0  1  0  5  4  1  0  1 | c = aux
 1  1  0 122  0  6  8  3  0  0 | d = determiner
 0  1  0  1 39  8  6  0  0  0 | e = invariant
41  2  1  5  3 563 11  1  0 14 | f = noun
 4  3  1 11  3 11 125  3  0  0 | g = preposition
 0  2  0  4  0  3  4 22  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 2  0  0  0  0 30  0  0  0 27 | j = verb
```

También el meta-algoritmo AdaBoostM1 basado en el algoritmo de árboles J48 ha empeorado su efectividad, alrededor de un 2% comparado con la primera fase.

### 7.2.1.11 Experimento 11

Para el Experimento 11 se ejecuta la clasificación con el meta-algoritmo **MultiBoostAB** basado en el **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.MultiBoostAB -C 3 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      963                75.117 %
Incorrectly Classified Instances    319                24.883 %
Kappa statistic                    0.6247
Mean absolute error                 0.0501
Root mean squared error             0.2107
Relative absolute error             35.4374 %
Root relative squared error         79.3358 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.254    0.026    0.531    0.254    0.343    0.808    adjective
      0.372    0.006    0.667    0.372    0.478    0.813    adverb
      0         0.001    0         0         0         0.709    aux
      0.915    0.024    0.827    0.915    0.869    0.974    determiner
      0.691    0.007    0.826    0.691    0.752    0.935    invariant
      0.897    0.301    0.749    0.897    0.816    0.878    noun
      0.758    0.023    0.824    0.758    0.79     0.91     preposition
      0.771    0.01     0.675    0.771    0.72     0.921    pronoun
      0         0         0         0         0         0.043    qualifier
      0.373    0.011    0.629    0.373    0.468    0.828    verb
Weighted Avg.  0.751    0.16     0.73     0.751    0.728    0.882
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
34  1  0  1  0  95  2  0  0  1 | a = adjective
 1 16  0  2  0 19  2  2  0  1 | b = adverb
 0  0  0  2  0  7  3  0  0  0 | c = aux
 0  0  0 129  0  5  4  3  0  0 | d = determiner
 0  1  0  1 38 11  4  0  0  0 | e = invariant
27  5  0  5  4 575 10  4  0 11 | f = noun
 2  1  1 12  4 15 122  4  0  0 | g = preposition
 0  0  0  4  0  3  1 27  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 0  0  0  0  0 37  0  0  0 22 | j = verb
```

Con este meta-algoritmo de boosting se ha igualado el resultado del AdaBoostM1 de esta misma fase, pero es otro de los algoritmos que decae en eficacia, en un 0,66% para ser más exactos.

### 7.2.1.12 Experimento 12

El meta-algoritmo de clasificación utilizado en el Experimento 12 es el **Bagging**, apoyado en el algoritmo **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      973                75.897 %
Incorrectly Classified Instances    309                24.103 %
Kappa statistic                    0.6393
Mean absolute error                 0.0691
Root mean squared error             0.1925
Relative absolute error             48.8701 %
Root relative squared error         72.4745 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.336    0.034    0.536    0.336    0.413    0.851    adjective
      0.349    0.005    0.714    0.349    0.469    0.854    adverb
      0         0.002    0         0         0         0.746    aux
      0.894    0.025    0.818    0.894    0.854    0.976    determiner
      0.691    0.01     0.76     0.691    0.724    0.926    invariant
      0.899    0.275    0.766    0.899    0.827    0.872    noun
      0.745    0.02     0.845    0.745    0.792    0.949    preposition
      0.771    0.01     0.675    0.771    0.72     0.913    pronoun
      0         0         0         0         0         0.498    qualifier
      0.441    0.009    0.703    0.441    0.542    0.87     verb
Weighted Avg.  0.759    0.147    0.742    0.759    0.741    0.892
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
45  0  0  1  0 87  0  0  0  1 | a = adjective
 2 15  0  1  1 19  2  2  0  1 | b = adverb
 0  0  0  2  0  7  3  0  0  0 | c = aux
 1  0  0 126  0  6  4  4  0  0 | d = determiner
 0  1  0  1 38 11  4  0  0  0 | e = invariant
30  5  1  6  3 576  8  3  0  9 | f = noun
 4  0  1 13  7 12 120  4  0  0 | g = preposition
 0  0  0  4  1  2  1 27  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 2  0  0  0  0 31  0  0  0 26 | j = verb
```

Con este meta-algoritmo se cierra la segunda fase de experimentación que tiene a su disposición únicamente el número de regla de desambiguación para la palabra que se trata de clasificar. El resultado máximo no ha llegado ni al 76% de acierto, ha empeorado.



### 7.2.1.13 Conclusiones sobre la segunda fase (F2-1R)

Los resultados de la segunda fase no han resultado satisfactorios. No se ha mejorado el valor conseguido en la primera fase de experimentación, por lo que parece que el contexto no está ayudando a desambiguar las palabras, ni tampoco las reglas elegidas. En la mayoría de los casos los algoritmos han experimentado pérdidas de efectividad.

De los meta-algoritmos, esta vez cabe destacar al Bagging con reglas Part. Debido a que la muestra es bastante pequeña, no se aprecia un aumento grande de carga computacional en la creación de los modelos de aprendizaje, y por ende, tampoco se consiguen elevados valores de acierto.

A continuación se muestra una gráfica comparando los porcentajes de acierto de los algoritmos utilizados en la segunda fase de experimentación con el número de regla de desambiguación solo para la palabra a desambiguar:

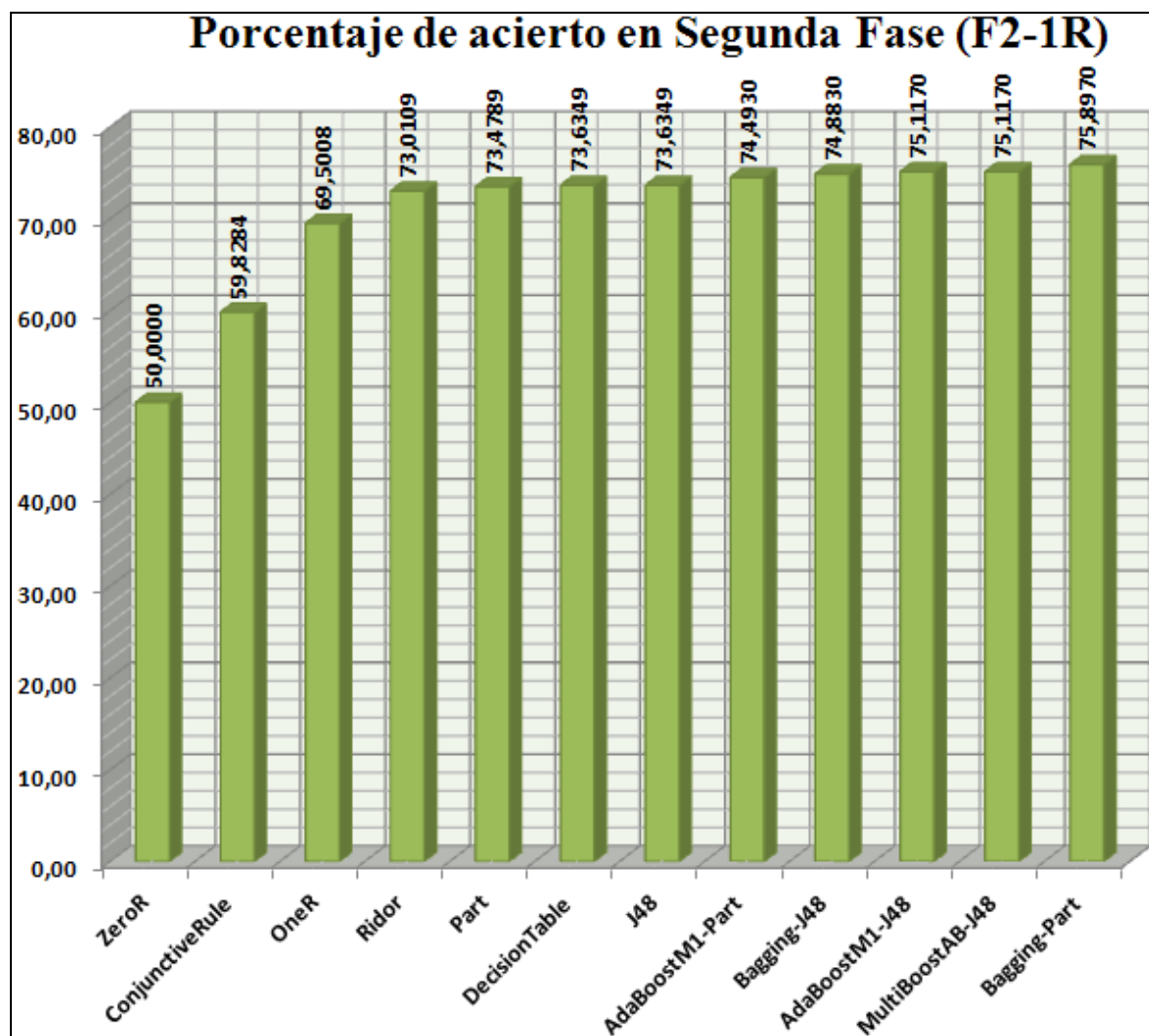


Figura 29: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con una regla

## 7.2.2 Experimentos con número de regla para la palabra y contexto (F2-3R)

### 7.2.2.1 Experimento 1

En el Experimento 1 se efectúa la clasificación con el algoritmo **ZeroR**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.ZeroR
Correctly Classified Instances      641          50      %
Incorrectly Classified Instances    641          50      %
Kappa statistic                     0
Mean absolute error                 0.1414
Root mean squared error             0.2656
Relative absolute error              100      %
Root relative squared error          100      %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
          0         0         0         0         0      0.489  adjective
          0         0         0         0         0      0.474  adverb
          0         0         0         0         0      0.433   aux
          0         0         0         0         0      0.496  determiner
          0         0         0         0         0      0.476  invariant
          1         1         0.5         1         0.667      0.499   noun
          0         0         0         0         0      0.497  preposition
          0         0         0         0         0      0.463  pronoun
          0         0         0         0         0       0.05  qualifier
          0         0         0         0         0      0.492   verb
Weighted Avg.    0.5      0.5      0.25      0.5      0.333      0.493
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix===
a  b  c  d  e  f  g  h  i  j  <-- classified as
0  0  0  0  0 134  0  0  0  0 | a = adjective
0  0  0  0  0  43  0  0  0  0 | b = adverb
0  0  0  0  0  12  0  0  0  0 | c = aux
0  0  0  0  0 141  0  0  0  0 | d = determiner
0  0  0  0  0  55  0  0  0  0 | e = invariant
0  0  0  0  0 641  0  0  0  0 | f = noun
0  0  0  0  0 161  0  0  0  0 | g = preposition
0  0  0  0  0  35  0  0  0  0 | h = pronoun
0  0  0  0  0  1  0  0  0  0 | i = qualifier
0  0  0  0  0  59  0  0  0  0 | j = verb
```

El resultado no mejora el porcentaje obtenido por el mismo algoritmo en la primera fase, ni en la segunda fase con el número de regla para la palabra a desambiguar, sólo lo iguala.

### 7.2.2.2 Experimento 2

Para el Experimento 2 se ejecuta la clasificación con el algoritmo **ConjunctiveRule**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Correctly Classified Instances      767          59.8284 %
Incorrectly Classified Instances    515          40.1716 %
Kappa statistic                    0.2691
Mean absolute error                 0.1202
Root mean squared error             0.2453
Relative absolute error             85.0017 %
Root relative squared error         92.3472 %
Total Number of Instances          1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0         0         0         0         0         0.554    adjective
      0         0         0         0         0         0.513    adverb
      0         0         0         0         0         0.422    aux
      0.929     0.028     0.804     0.929     0.862     0.943    determiner
      0         0         0         0         0         0.525    invariant
      0.992     0.754     0.568     0.992     0.723     0.62     noun
      0         0         0         0         0         0.505    preposition
      0         0         0         0         0         0.479    pronoun
      0         0         0         0         0         0.107    qualifier
      0         0         0         0         0         0.561    verb
Weighted Avg.  0.598     0.38     0.373     0.598     0.456     0.618

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
0  0  0  1  0 133  0  0  0  0 | a = adjective
0  0  0  2  0  41  0  0  0  0 | b = adverb
0  0  0  2  0  10  0  0  0  0 | c = aux
0  0  0 131  0  10  0  0  0  0 | d = determiner
0  0  0  2  0  53  0  0  0  0 | e = invariant
0  0  0  5  0 636  0  0  0  0 | f = noun
0  0  0 17  0 144  0  0  0  0 | g = preposition
0  0  0  3  0  32  0  0  0  0 | h = pronoun
0  0  0  0  0  1  0  0  0  0 | i = qualifier
0  0  0  0  0  59  0  0  0  0 | j = verb

```

El valor del resultado es el mismo que el obtenido para una sola regla de desambiguación. Los valores sólo se están clasificando en sustantivos (*noun*) y determinantes (*determiner*).

### 7.2.2.3 Experimento 3

El algoritmo de clasificación utilizado en el Experimento 3 es el llamado **OneR**.

Los resultados de la clasificación son:

```
Scheme: weka.classifiers.rules.OneR -B 6
Correctly Classified Instances      891                69.5008 %
Incorrectly Classified Instances    391                30.4992 %
Kappa statistic                    0.5248
Mean absolute error                 0.061
Root mean squared error             0.247
Relative absolute error             43.1406 %
Root relative squared error         92.9961 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.157    0.027    0.404    0.157    0.226    0.565    adjective
      0.209    0.005    0.6    0.209    0.31    0.602    adverb
      0.083    0.005    0.143    0.083    0.105    0.539    aux
      0.801    0.042    0.702    0.801    0.748    0.88    determiner
      0.545    0.007    0.769    0.545    0.638    0.769    invariant
      0.906    0.385    0.702    0.906    0.791    0.761    noun
      0.702    0.032    0.758    0.702    0.729    0.835    preposition
      0.429    0    1    0.429    0.6    0.714    pronoun
      0    0    0    0    0    0.5    qualifier
      0.136    0.007    0.5    0.136    0.213    0.565    verb
Weighted Avg.    0.695    0.205    0.67    0.695    0.658    0.745
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
21  0  0  3  0 109  1  0  0  0 | a = adjective
 1  9  0  2  1  24  6  0  0  0 | b = adverb
 1  0  1  0  1  9  0  0  0  0 | c = aux
 1  0  0 113  0  17  10  0  0  0 | d = determiner
 1  0  0  2  30  19  3  0  0  0 | e = invariant
20  2  4  14  2 581  12  0  0  6 | f = noun
 2  3  1  20  5  17 113  0  0  0 | g = preposition
 1  1  0  6  0  8  2  15  0  2 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 4  0  1  1  0  43  2  0  0  8 | j = verb
```

Al igual que sucede con el algoritmo anterior, se vuelve a clavar el resultado obtenido con OneR para una sola regla de desambiguación.

### 7.2.2.4 Experimento 4

En el Experimento 4 se efectúa la clasificación con el meta-algoritmo **AdaBoostM1** sobre el algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      913                71.2168 %
Incorrectly Classified Instances    369                28.7832 %
Kappa statistic                    0.5883
Mean absolute error                 0.0578
Root mean squared error             0.2257
Relative absolute error             40.9122 %
Root relative squared error         85.0025 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.403    0.059    0.443    0.403    0.422    0.791    adjective
      0.419    0.015    0.486    0.419    0.45    0.838    adverb
      0         0.005    0         0         0         0.806    aux
      0.801    0.02    0.831    0.801    0.816    0.972    determiner
      0.655    0.013    0.692    0.655    0.673    0.952    invariant
      0.799    0.228    0.778    0.799    0.788    0.86    noun
      0.801    0.038    0.75    0.801    0.775    0.917    preposition
      0.657    0.011    0.622    0.657    0.639    0.918    pronoun
      0         0.002    0         0         0         0.08    qualifier
      0.475    0.026    0.467    0.475    0.471    0.82    verb
Weighted Avg.    0.712    0.13    0.705    0.712    0.708    0.874
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
54  0  0  1  0  72  2  0  0  5 |  a = adjective
 1 18  0  1  1 15  4  2  0  1 |  b = adverb
 0  0  0  0  1  5  4  1  0  1 |  c = aux
 0  1  1 113  2  7 10  7  0  0 |  d = determiner
 0  1  0  1 36  9  7  1  0  0 |  e = invariant
61 12  1  7  8 512 13  1  2 24 |  f = noun
 2  4  2  9  3  9 129  2  0  1 |  g = preposition
 1  1  1  4  1  1  3 23  0  0 |  h = pronoun
 0  0  0  0  0  1  0  0  0  0 |  i = qualifier
 3  0  1  0  0 27  0  0  0 28 |  j = verb
```

Esta vez, el meta-algoritmo AdaBoostM1 ejecutado sobre J48 ha empeorado el valor de su porcentaje de éxito en casi un 4% con respecto al resultado obtenido para una sola regla de desambiguación.

### 7.2.2.5 Experimento 5

Para el Experimento 5 se ejecuta la clasificación con el algoritmo **Ridor**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.rules.Ridor -F 3 -S 1 -N 2.0
Correctly Classified Instances      925                72.1529 %
Incorrectly Classified Instances    357                27.8471 %
Kappa statistic                    0.581
Mean absolute error                 0.0557
Root mean squared error             0.236
Relative absolute error             39.3892 %
Root relative squared error         88.8608 %
Total Number of Instances          1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.291    0.03    0.527    0.291    0.375     0.63    adjective
      0.349    0.015   0.441    0.349    0.39     0.667   adverb
      0.333    0.006   0.364    0.333    0.348     0.664    aux
      0.794    0.033   0.747    0.794    0.77     0.881  determiner
      0.582    0.015   0.627    0.582    0.604     0.783  invariant
      0.891    0.301   0.747    0.891    0.813     0.795   noun
      0.758    0.022    0.83    0.758    0.792     0.868  preposition
      0.429    0.006   0.652    0.429    0.517     0.711  pronoun
      0        0        0        0        0        0.5    qualifier
      0.254    0.011   0.536    0.254    0.345     0.622   verb
Weighted Avg.  0.722    0.162   0.703    0.722    0.702     0.78

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
39  1  0  4  2 85  1  1  0  1 | a = adjective
 2 15  0  4  3 14  4  1  0  0 | b = adverb
 0  1  4  1  1  4  1  0  0  0 | c = aux
 1  0  0 112  5 14  6  2  0  1 | d = determiner
 1  2  0  4 32 14  2  0  0  0 | e = invariant
28  7  5  5  3 571 10  1  0 11 | f = noun
 1  5  1 11  5 13 122  3  0  0 | g = preposition
 0  3  0  7  0  9  1 15  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 2  0  1  2  0 39  0  0  0 15 | j = verb

```

Se ha experimentado un empeoramiento con respecto al resultado de Ridor en la segunda fase con una regla de desambiguación de más de 0,9%, por lo que se puede decir que no ha sido de provecho el añadir las reglas para el contexto.

### 7.2.2.6 Experimento 6

El algoritmo de clasificación utilizado en el Experimento 6 es el **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.PART -M 2 -C 0.25 -Q 1
Correctly Classified Instances      926                72.2309 %
Incorrectly Classified Instances    356                27.7691 %
Kappa statistic                    0.5969
Mean absolute error                 0.0671
Root mean squared error             0.2111
Relative absolute error             47.4853 %
Root relative squared error         79.5027 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.388    0.053    0.46    0.388    0.421    0.779    adjective
      0.442    0.018    0.463    0.442    0.452    0.788    adverb
      0        0.008    0        0        0        0.639    aux
      0.851    0.019    0.845    0.851    0.848    0.953    determiner
      0.636    0.015    0.648    0.636    0.642    0.875    invariant
      0.833    0.245    0.773    0.833    0.802    0.83     noun
      0.745    0.028    0.795    0.745    0.769    0.909    preposition
      0.629    0.009    0.667    0.629    0.647    0.857    pronoun
      0        0        0        0        0        0.498    qualifier
      0.407    0.019    0.511    0.407    0.453    0.798    verb
Weighted Avg.    0.722    0.136    0.712    0.722    0.716    0.846
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
52  0  1  1  0  77  1  0  0  2 | a = adjective
 2 19  0  0  1 18  1  1  0  1 | b = adverb
 0  0  0  2  0  6  3  0  0  1 | c = aux
 3  1  2 120  0  6  7  2  0  0 | d = determiner
 1  1  0  1 35 10  7  0  0  0 | e = invariant
43 11  4  5 14 534  8  3  0 19 | f = noun
 7  6  2 11  3  7 120  5  0  0 | g = preposition
 0  3  1  2  1  4  2 22  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 5  0  0  0  0 28  2  0  0 24 | j = verb
```

Con este algoritmo también se ha experimentado una pérdida de efectividad de casi 1,25% con respecto al mismo clasificador, pero para una sola regla de desambiguación. Hasta ahora, el contar con las reglas del contexto no está suponiendo ninguna mejora.

## 7.2.2.7 Experimento 7

En el Experimento 7 se efectúa la clasificación con el algoritmo **DecisionTable**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Correctly Classified Instances          935                72.9329 %
Incorrectly Classified Instances        347                27.0671 %
Kappa statistic                        0.5823
Mean absolute error                    0.0959
Root mean squared error                0.205
Relative absolute error                 67.8314 %
Root relative squared error            77.1807 %
Total Number of Instances             1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.119    0.024    0.372    0.119    0.181    0.829    adjective
      0.302    0.015    0.419    0.302    0.351    0.831    adverb
      0.417    0.005    0.455    0.417    0.435    0.923    aux
      0.943    0.031    0.792    0.943    0.861    0.975    determiner
      0.6      0.013    0.673    0.6      0.635    0.928    invariant
      0.925    0.346    0.728    0.925    0.815    0.841    noun
      0.714    0.013    0.885    0.714    0.79     0.967    preposition
      0.486    0.002    0.85     0.486    0.618    0.956    pronoun
      0        0        0        0        0        0.933    qualifier
      0.169    0.004    0.667    0.169    0.27     0.808    verb
Weighted Avg. 0.729    0.182    0.702    0.729    0.693    0.876
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
16  0  0  1  0 115  1  0  0  1 | a = adjective
 2 13  0  2  1 22  3  0  0  0 | b = adverb
 1  0  5  2  0  3  1  0  0  0 | c = aux
 2  0  0 133  1  3  0  2  0  0 | d = determiner
 0  1  0  3 33 15  3  0  0  0 | e = invariant
13 11  5  5  7 593  2  1  0  4 | f = noun
 6  5  0 17  5 13 115  0  0  0 | g = preposition
 1  0  0  5  2  5  5 17  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 2  1  1  0  0 45  0  0  0 10 | j = verb
```

Con el algoritmo también se ha experimentado una caída de efectividad de 0,7% con respecto a la clasificación del DecisionTable con una sola regla de desambiguación, de momento no hay ninguna mejora.



### 7.2.2.8 Experimento 8

Para el Experimento 8 se ejecuta la clasificación con el meta-algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Correctly Classified Instances      935                72.9329 %
Incorrectly Classified Instances    347                27.0671 %
Kappa statistic                    0.5928
Mean absolute error                 0.0769
Root mean squared error             0.2075
Relative absolute error             54.3994 %
Root relative squared error         78.1475 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.269    0.033    0.486    0.269    0.346    0.756    adjective
      0.233    0.006    0.556    0.233    0.328    0.737    adverb
      0        0.001    0        0        0        0.653    aux
      0.908    0.029    0.795    0.908    0.848    0.963    determiner
      0.655    0.013    0.692    0.655    0.673    0.824    invariant
      0.886    0.307    0.742    0.886    0.808    0.801    noun
      0.665    0.02    0.829    0.665    0.738    0.912    preposition
      0.8      0.014    0.622    0.8      0.7      0.906    pronoun
      0        0        0        0        0        0.498    qualifier
      0.373    0.012    0.595    0.373    0.458    0.802    verb
Weighted Avg.  0.729    0.164    0.706    0.729    0.706    0.828
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
36  0  0  1  0 94  1  0  0  2 | a = adjective
 0 10  0  2  1 26  1  2  0  1 | b = adverb
 0  0  0  2  0  7  3  0  0  0 | c = aux
 1  0  0 128  0  4  4  4  0  0 | d = determiner
 0  0  0  2 36 12  5  0  0  0 | e = invariant
30  4  0  6 10 568  7  4  0 12 | f = noun
 6  4  1 17  5 14 107  7  0  0 | g = preposition
 0  0  0  3  0  3  1 28  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 1  0  0  0  0 36  0  0  0 22 | j = verb
```

Al igual que ha ocurrido con los experimentos que solo contaban con una regla de desambiguación, J48 y DecisionTable también coinciden en resultados en esta fase, lo cual no es una buena noticia porque tampoco han aumentado el valor del porcentaje.

### 7.2.2.9 Experimento 9

El meta-algoritmo de clasificación utilizado en el Experimento 9 es el **AdaBoostM1**, apoyado en el algoritmo **Part**.

Los resultados de la clasificación son:

```

Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances          943              73.5569 %
Incorrectly Classified Instances        339              26.4431 %
Kappa statistic                        0.6168
Mean absolute error                    0.0536
Root mean squared error                0.2152
Relative absolute error                 37.9275 %
Root relative squared error            81.0442 %
Total Number of Instances             1282

```

La tabla de precisión detallada por clases resultante:

```

=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.381    0.054    0.451    0.381    0.413     0.821   adjective
      0.395    0.011    0.548    0.395    0.459     0.833   adverb
      0         0.003    0         0         0         0.822   aux
      0.851    0.021    0.833    0.851    0.842     0.979   determiner
      0.655    0.011    0.72     0.655    0.686     0.95    invariant
      0.838    0.231    0.784    0.838    0.81      0.88    noun
      0.795    0.031    0.785    0.795    0.79      0.93    preposition
      0.686    0.012    0.615    0.686    0.649     0.955   pronoun
      0         0.002    0         0         0         0.919   qualifier
      0.508    0.017    0.588    0.508    0.545     0.849   verb
Weighted Avg. 0.736    0.129    0.723    0.736    0.728     0.892

```

La matriz de confusión del experimento es:

```

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
51  1  0  1  0  76  2  0  0  3 | a = adjective
 2 17  0  1  1 18  3  1  0  0 | b = adverb
 0  1  0  1  1  5  3  0  0  1 | c = aux
 1  1  0 120  0  6  7  6  0  0 | d = determiner
 1  0  0  1 36  6  8  3  0  0 | e = invariant
51  7  2  6  7 537  9  3  2 17 | f = noun
 2  2  1 10  4 12 128  2  0  0 | g = preposition
 0  2  1  4  1  1  2 24  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 5  0  0  0  0 23  1  0  0 30 | j = verb

```

En este caso, el algoritmo Part ha conseguido mejorar, aunque sea en muy poca cantidad, el grado de éxito que se consiguió en la segunda fase con una sola regla de desambiguación. El incremento ha sido 0,08%.

### 7.2.2.10 Experimento 10

En el Experimento 10 se efectúa la clasificación con el meta-algoritmo **Bagging**, tomando como base el algoritmo **Part**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.rules.PART -- -M 2 -C 0.25 -Q 1
Correctly Classified Instances      954                74.415 %
Incorrectly Classified Instances    328                25.585 %
Kappa statistic                    0.6205
Mean absolute error                 0.0671
Root mean squared error             0.1917
Relative absolute error             47.4456 %
Root relative squared error         72.1833 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.366    0.039    0.521    0.366    0.43       0.853    adjective
      0.326    0.012    0.483    0.326    0.389     0.828    adverb
      0         0.001    0         0         0         0.781    aux
      0.872    0.024    0.82     0.872    0.845     0.981    determiner
      0.655    0.012    0.706    0.655    0.679     0.928    invariant
      0.875    0.27     0.764    0.875    0.816     0.883    noun
      0.758    0.022    0.83     0.758    0.792     0.956    preposition
      0.686    0.01     0.649    0.686    0.667     0.903    pronoun
      0         0         0         0         0         0.498    qualifier
      0.424    0.011    0.641    0.424    0.51      0.863    verb
Weighted Avg.   0.744    0.146    0.725    0.744    0.729     0.898
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
49  0  0  1  0  80  1  0  0  3 |  a = adjective
 1 14  0  2  0  20  3  2  0  1 |  b = adverb
 0  1  0  2  0  6  3  0  0  0 |  c = aux
 0  2  0 123  0  8  4  4  0  0 |  d = determiner
 0  1  0  1 36 12  5  0  0  0 |  e = invariant
39  8  0  5  9 561  6  3  0 10 |  f = noun
 2  2  1 12  5 13 122  4  0  0 |  g = preposition
 0  1  0  4  1  2  3 24  0  0 |  h = pronoun
 0  0  0  0  0  1  0  0  0  0 |  i = qualifier
 3  0  0  0  0 31  0  0  0 25 |  j = verb
```

También el meta-algoritmo Bagging basado en el algoritmo de reglas Part ha empeorado su efectividad, alrededor de un 1,5% comparado con la segunda fase con una sola regla de desambiguación.

### 7.2.2.11 Experimento 11

Para el Experimento 11 se ejecuta la clasificación con el meta-algoritmo **Bagging** basado en el **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      956                74.571 %
Incorrectly Classified Instances    326                25.429 %
Kappa statistic                    0.6163
Mean absolute error                 0.0716
Root mean squared error             0.1951
Relative absolute error             50.6207 %
Root relative squared error         73.4531 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.291    0.027    0.557    0.291    0.382    0.858    adjective
      0.186    0.005    0.571    0.186    0.281    0.834    adverb
      0         0.001    0         0         0         0.91    aux
      0.915    0.029    0.796    0.915    0.851    0.973    determiner
      0.673    0.012    0.712    0.673    0.692    0.927    invariant
      0.902    0.301    0.75     0.902    0.819    0.863    noun
      0.689    0.014    0.874    0.689    0.771    0.943    preposition
      0.771    0.013    0.628    0.771    0.692    0.908    pronoun
      0         0         0         0         0         0.498    qualifier
      0.458    0.012    0.643    0.458    0.535    0.898    verb
Weighted Avg. 0.746    0.16    0.727    0.746    0.722    0.889
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
39  0  0  1  0 92  1  0  0  1 | a = adjective
 0  8  0  2  1 28  1  2  0  1 | b = adverb
 0  0  0  2  0  7  3  0  0  0 | c = aux
 1  0  0 129  0  4  3  4  0  0 | d = determiner
 0  1  0  1 37 11  4  1  0  0 | e = invariant
24  4  0  6  9 578  3  4  0 13 | f = noun
 5  1  1 17  4 17 111  5  0  0 | g = preposition
 0  0  0  4  1  2  1 27  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 1  0  0  0  0 31  0  0  0 27 | j = verb
```

Con este meta-algoritmo de bagging basado en el algoritmo J48, se han conseguido peores resultados, concretamente más de un 0,3%, que en la segunda fase con una sola regla de desambiguación.

### 7.2.2.12 Experimento 12

El meta-algoritmo de clasificación utilizado en el Experimento 12 es el **MultiBoostAB**, apoyado en el algoritmo **J48**.

Los resultados de la clasificación son:

```
Scheme:weka.classifiers.meta.MultiBoostAB -C 3 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Correctly Classified Instances      957                74.649 %
Incorrectly Classified Instances    325                25.351 %
Kappa statistic                    0.6258
Mean absolute error                 0.0501
Root mean squared error             0.2135
Relative absolute error             35.4597 %
Root relative squared error         80.3939 %
Total Number of Instances          1282
```

La tabla de precisión detallada por clases resultante:

```
=== Detailed Accuracy By Class ===
      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.328    0.038     0.5         0.328    0.396     0.768  adjective
      0.349    0.013     0.484       0.349    0.405     0.823  adverb
      0         0.001     0         0         0         0.722  aux
      0.908    0.024     0.826       0.908    0.865     0.976  determiner
      0.673    0.011     0.725       0.673    0.698     0.95   invariant
      0.871    0.259     0.771       0.871    0.818     0.873  noun
      0.783    0.022     0.834       0.783    0.808     0.922  preposition
      0.657    0.007     0.719       0.657    0.687     0.918  pronoun
      0         0         0         0         0         0.124  qualifier
      0.441    0.019     0.531       0.441    0.481     0.839  verb
Weighted Avg.  0.746    0.141     0.725       0.746    0.731     0.879
```

La matriz de confusión del experimento es:

```
=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  <-- classified as
44  0  0  1  0 84  1  0  0  4 | a = adjective
 0 15  0  2  2 19  2  2  0  1 | b = adverb
 0  0  0  2  0  6  4  0  0  0 | c = aux
 1  1  0 128  1  4  3  3  0  0 | d = determiner
 0  2  0  2 37  9  4  1  0  0 | e = invariant
38  8  0  5  7 558  6  1  0 18 | f = noun
 2  4  0 12  4 11 126  2  0  0 | g = preposition
 0  1  1  3  0  2  5 23  0  0 | h = pronoun
 0  0  0  0  0  1  0  0  0  0 | i = qualifier
 3  0  0  0  0 30  0  0  0 26 | j = verb
```

Con este meta-algoritmo se cierra la segunda fase de experimentación que tiene a su disposición los números de regla de desambiguación para la palabra que se trata de clasificar y los de su contexto. El resultado máximo no ha llegado ni al 75% de acierto.

### 7.2.2.13 Conclusiones sobre la segunda fase (F2-3R)

Los resultados de la segunda fase en la que se incluían los números de las reglas para el contexto han sido peores aún que los de los que no disponían del número de las reglas. No solo no se ha mejorado el valor conseguido en la primera fase de experimentación, sino que se ha empeorado el que ya había. En la mayoría de los casos los algoritmos han experimentado pérdidas de efectividad, salvo en el Part, que ha mejorado levemente.

De los meta-algoritmos, esta vez cabe destacar al Bagging basado en árboles de decisión J48, que se muestra como el más fuerte. Debido a que la muestra es bastante pequeña, no se aprecia un aumento grande de carga computacional en la creación de los modelos de aprendizaje, y por ende, tampoco se consiguen elevados valores de acierto.

A continuación se muestra una gráfica comparando los porcentajes de acierto de los algoritmos utilizados en la segunda fase de experimentación con reglas de desambiguación para la palabra a clasificar y su contexto:

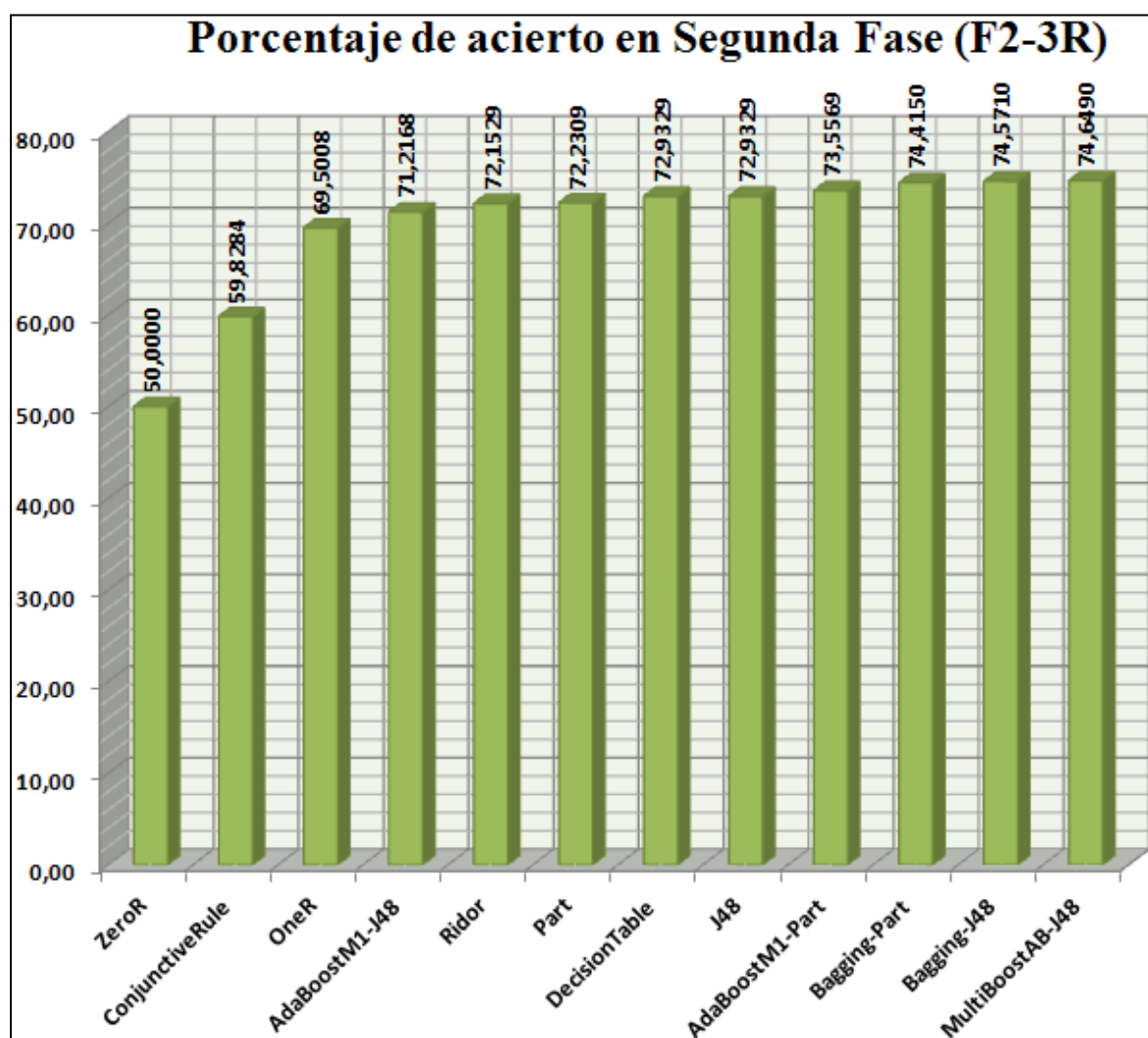


Figura 30: Gráfica de comparación de los algoritmos en la 2ª fase de experimentación con tres reglas

## 8. RESULTADOS

Tras la finalización de la fase de experimentación, se pueden sacar bastantes conclusiones a partir de los resultados. La idea principal es que, para efectuar una desambiguación efectiva, la muestra de datos debe ser lo mayor posible, al igual que el diccionario contra el que se contrasta. A continuación se muestra una gráfica en la que se comparan los datos de las experimentaciones.

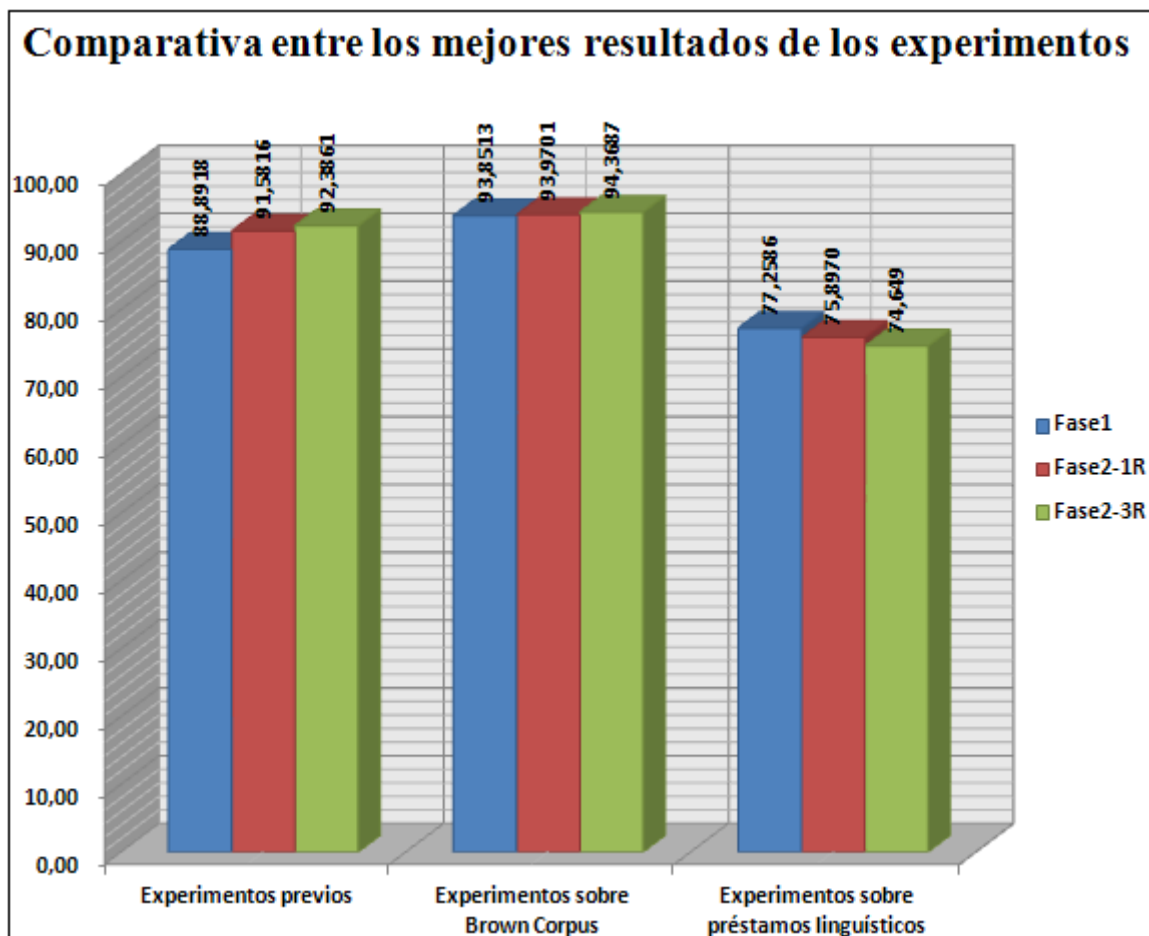


Figura 31: Gráfica de comparación entre experimentos realizados

Tanto en los experimentos previos como en los realizados sobre la totalidad del Brown Corpus se observa un incremento progresivo del éxito a medida que se van añadiendo reglas de desambiguación y se atiende al contexto de las palabras. La diferencia del porcentaje de aciertos entre ambos tipos de experimentos radica, principalmente, en que en los previos la muestra de datos era casi el 10% del tamaño total del Brown Corpus, y además el diccionario de datos utilizado era relativamente limitado, no contaba con la totalidad de las palabras que aparecen en el corpus. Otra de las diferencias se encuentra en las categorías a desambiguar, ya que en los experimentos realizados sobre la totalidad del corpus se aislaron tanto los vocablos extranjeros como los signos de puntuación y demás

términos que no estuvieran constituidos exclusivamente por caracteres alfabéticos. En los experimentos previos se trató el texto íntegro, sin eliminar ningún token. Aún así, se puede apreciar que tanto en un caso como en otro, el grado de acierto se incrementa cuando interviene el contexto. Cabe reseñar que en los experimentos previos no se utilizó ningún meta-algoritmo, que son los que mayor capacidad de acierto han mostrado cuando se ha recurrido a ellos. Es cierto que es más acentuado el progreso en los experimentos previos, ya que en la primera fase no se llega ni al 89% de instancias correctas, por lo que añadir el contexto supone una gran inyección de información valiosa. En los experimentos sobre la totalidad del corpus, la diferencia máxima entre la primera y la segunda fase no llega ni al 1%, pero es que se trata de casi un millón de palabras a clasificar, y arañar unas décimas supone un avance muy grande.

Justamente todo lo contrario ocurre con los experimentos realizados sobre las palabras extranjeras encontradas en el corpus Brown. Esto se debe, en gran medida, a que son palabras de distintas lenguas y que a veces aparecen dispersas por el texto, lo que hace imposible poder desambiguarlas. Si ya resulta complicado a veces desambiguar palabras dentro de textos de una sola lengua, intentar el proceso sobre palabras de diferentes idiomas a la vez se antoja extremadamente problemático, de ahí el bajo éxito de los experimentos. También se ha de tener en cuenta que ni el contexto está presente de manera correcta (por la dispersión de las palabras, muchas veces no se encuentran frases completas, sólo vocablos aislados), ni la muestra es lo suficientemente grande como para llegar a un alto grado de fiabilidad, ni el diccionario de datos es homogéneo (conviven palabras de distintos idiomas). Todo esto puede ser la razón que explique cómo, de una fase a otra, cuando se supone que al añadir más información relevante se debe incrementar el grado de acierto, se vaya perdiendo eficacia.

A continuación se muestra una tabla que resume todos los resultados de los experimentos:

Tipo Experimento	Fase	Instancias	Algoritmo utilizado	Porcentaje de acierto
Experimentos previos	Fase 1	100817	ZeroR	30,3143
			ConjunctiveRules	39,7463
			OneR	79,1315
			Ridor	87,5091
			DecisionTable	88,2629
			J48	88,8461
			<b>Part</b>	<b>88,8918</b>
	Fase 2-1R	100815	ZeroR	30,3149
			ConjunctiveRules	40,6874
			OneR	89,6285
			DecisionTable	90,0342
			Ridor	90,4826
			J48	91,5409
			<b>Part</b>	<b>91,5816</b>



Tipo Experimento	Fase	Instancias	Algoritmo utilizado	Porcentaje de acierto
Experimentos previos	Fase 2-3R	100815	ZeroR	30,3149
			ConjunctiveRules	40,6874
			OneR	89,6285
			DecisionTable	90,0551
			Ridor	90,2594
			Part	92,0210
			<b>J48</b>	<b>92,3861</b>
Experimentos realizados sobre Brown Corpus	Fase 1	990448	ZeroR	26,1246
			ConjunctiveRule	35,0777
			Ridor	93,3194
			OneR	93,3989
			DecisionTable	93,5255
			Part	93,6484
			J48	93,7329
			Bagging-J48	93,7479
			MultiBoostAB-J48	93,7620
			Bagging-Part	93,7719
			AdaBoostM1-Part	93,8085
			<b>AdaBoostM1-J48</b>	<b>93,8513</b>
	Fase 2-1R	990446	ZeroR	26,1245
			ConjunctiveRule	39,1258
			OneR	91,1708
			DecisionTable	93,1482
			Ridor	93,6915
			Part	93,7086
			J48	93,8447
			Bagging-J48	93,8569
			Bagging-Part	93,8845
			MultiBoostAB-J48	93,8864
			AdaBoostM1-Part	93,9285
			<b>AdaBoostM1-J48</b>	<b>93,9701</b>
	Fase 2-3R	990446	ZeroR	26,1245
			ConjunctiveRule	39,1258
			OneR	91,1708
			DecisionTable	93,1482
			Ridor	93,9316
			Part	93,9616
			J48	94,0187
			Bagging-J48	94,0448
			MultiBoostAB-J48	94,0895
			Bagging-Part	94,1166
			AdaBoostM1-Part	94,2907
			<b>AdaBoostM1-J48</b>	<b>94,3687</b>
Experimentos realizados sobre préstamos lingüísticos encontrados en Brown Corpus	Fase 1	1284	ZeroR	50,0000
			ConjunctiveRule	58,0997
			OneR	69,1589
			DecisionTable	70,3271
			Ridor	70,8723
			Part	74,7664
			Bagging-J48	75,3894
			MultiBoostAB-J48	75,7788
			J48	76,0125
			Bagging-Part	76,5576

Tipo Experimento	Fase	Instancias	Algoritmo utilizado	Porcentaje de acierto
Experimentos realizados sobre préstamos lingüísticos encontrados en Brown Corpus	Fase 1	1284	AdaBoostM1-J48	77,1807
			<b>AdaBoostM1-Part</b>	<b>77,2586</b>
	Fase 2-1R	1282	ZeroR	50,0000
			ConjunctiveRule	59,8284
			OneR	69,5008
			Ridor	73,0109
			Part	73,4789
			DecisionTable	73,6349
			J48	73,6349
			AdaBoostM1-Part	74,4930
			Bagging-J48	74,8830
			AdaBoostM1-J48	75,1170
			MultiBoostAB-J48	75,1170
			<b>Bagging-Part</b>	<b>75,8970</b>
	Fase 2-3R	1282	ZeroR	50,0000
			ConjunctiveRule	59,8284
			OneR	69,5008
			AdaBoostM1-J48	71,2168
			Ridor	72,1529
			Part	72,2309
			DecisionTable	72,9329
			J48	72,9329
			AdaBoostM1-Part	73,5569
			Bagging-Part	74,4150
			Bagging-J48	74,5710
			<b>MultiBoostAB-J48</b>	<b>74,6490</b>

Tabla 15: Resultados de los experimentos

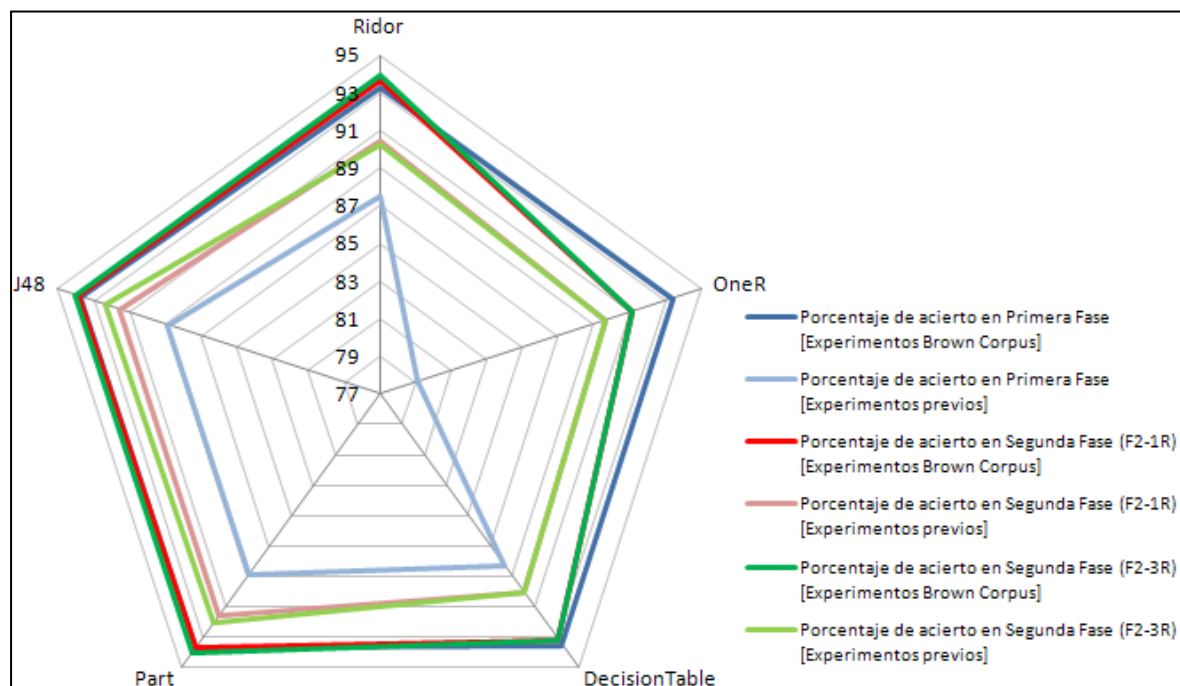


Figura 32: Comparativa de los algoritmos con mayor acierto en los experimentos previos con los mismos realizados sobre la totalidad de Brown Corpus

## 9. PLANIFICACIÓN

Para afrontar un proyecto informático con garantías y alcanzar el objetivo marcado, siempre se debe realizar una planificación. Es durante la planificación donde se plantean los tres cimientos sobre los que se apoyan el correcto desarrollo del proyecto y su viabilidad. El principal es la **calidad**, que vendrá dada por unas especificaciones concisas y claras, a las que se tendrá que ajustar perfectamente el producto software final. Otra son los **costes**, efectuando una estimación para presupuestar el impacto económico que ocasionaría afrontar el desarrollo. Y, por último, la **duración** del proyecto, que debe estar definida antes del comienzo para poder estimar si los tiempos de las tareas que componen el proceso de desarrollo son asumibles.

A continuación se pasa a detallar las estimaciones de tiempo y costes para este PFC, ya que de las especificaciones no hay nada que exponer.

### 9.1 Estimación de tiempo

Para estimar el tiempo que llevará completar el proyecto, lo primero que debe hacerse es dividirlo en las tareas que lo componen. Se puede dar el caso de que dos o más tareas coincidan en el tiempo, o que el comienzo de unas dependa de la conclusión de otras.

En este caso, como en todo proyecto software, se ha dividido en estas tareas principales:

- Toma de requisitos
- Análisis
- Diseño
- Implementación
- Pruebas (experimentación en este caso)
- Documentación

Las tareas principales se han podido dividir en subtareas, y éstas en actividades, las cuales ya cuentan con una estimación de tiempo aproximada por separado. Se ha utilizado un diagrama de Gantt<sup>4</sup> para expresar visualmente el alcance y detalle de las tareas, subtareas y actividades que componen este proyecto.

---

<sup>4</sup> Herramienta de gráficos de tiempos, creada por Henry L.Gantt en 1917, que resultan bastante eficaces para la planificación y la evaluación del avance de los proyectos. Cada barra que compone el gráfico simboliza una tarea del proyecto, y el eje horizontal representa el tiempo, de manera que se puede ilustrar el solapamiento de actividades durante el desarrollo del proyecto.

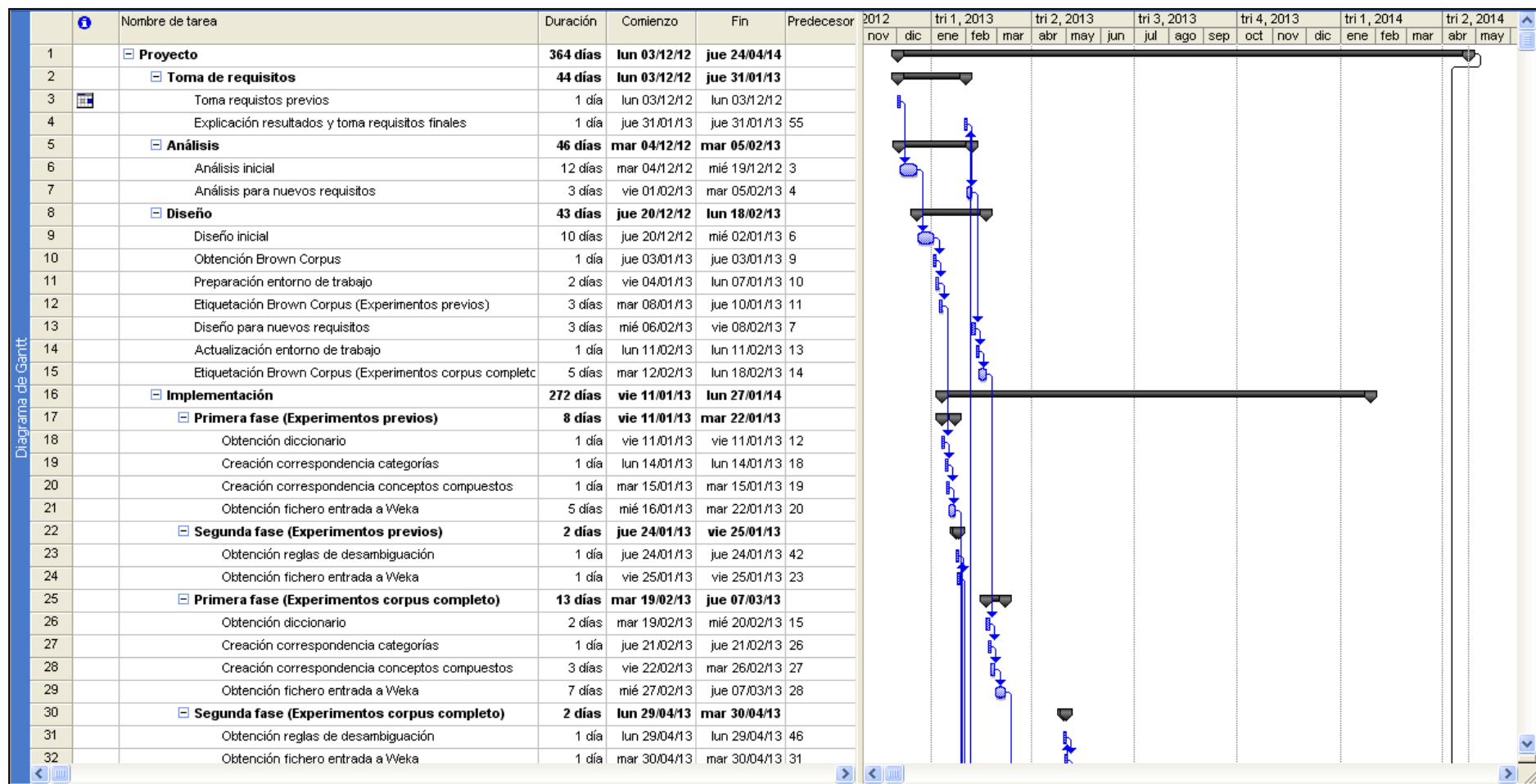


Figura 33: Diagrama de Gantt, Parte 1

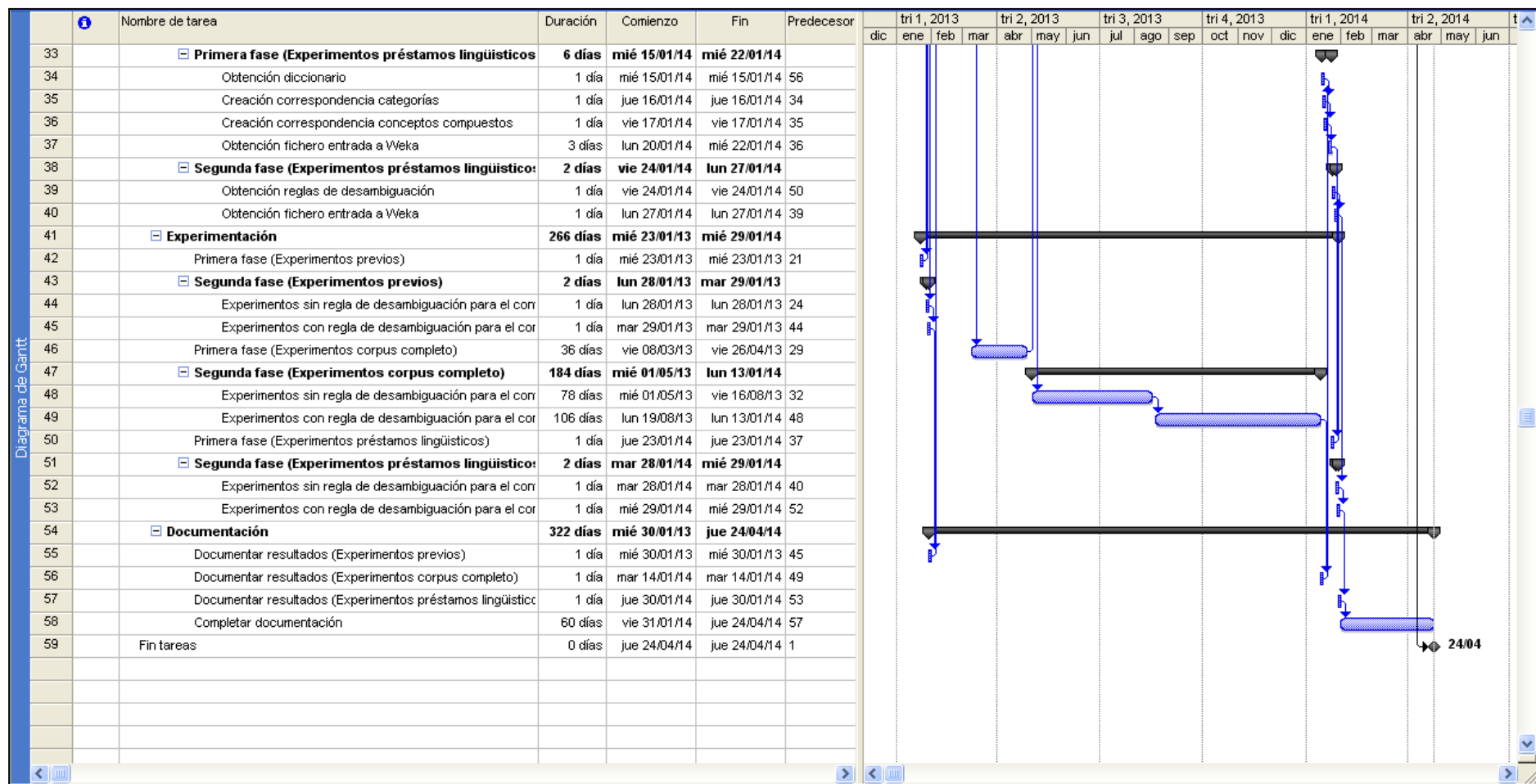


Figura 34: Diagrama de Gantt, Parte 2

## 9.2 Estimación de costes

Todo proyecto tiene un coste asociado a los recursos utilizados en el mismo. En este caso, dichos recursos han sido humanos y materiales. A continuación, se muestra un desglose del coste total del proyecto.

- Recursos Humanos

Para hacer frente al proyecto se ha dispuesto de un recurso con perfil analista programador, con una dedicación del 100% a lo largo de una semana laboral de 5 días y jornadas de 8 horas, de lunes a viernes. El sueldo mensual de este recurso es de 1.900 € netos, aproximadamente unos 10,80 €/h de trabajo. Como el proyecto ha tenido una duración de 16,9 meses, el coste de este recurso ha ascendido a 32.110 €.

- Recursos Materiales

Se han dispuesto de varios recursos materiales para la elaboración de este proyecto.

El principal recurso es un ordenador portátil con las siguientes características:

Dell Alienware M14x  
Procesador: Intel® Core™ i7 3630QM (6 MB Cache, hasta 3,4 GHz)  
Sistema operativo: Microsoft Windows XP Professional SP3  
Pantalla: WLED de alta definición (1600 x 900) y 14,0”  
Memoria Principal: 8 GB Dual Channel SDRAM DDR3 a 1600 MHz  
Memoria Secundaria: 500 GB SATA a 7200 rpm  
Tarjeta gráfica: GDDR5 NVIDIA® GeForce® GT 650M de 2 GB  
Unidad óptica: DVD+/-RW (lectura y escritura de DVD y CD)

Como se puede apreciar por sus características, es un ordenador muy potente. Se ha utilizado este PC por la gran carga de trabajo y procesamiento que requería el proyecto, sobre todo a nivel de memoria principal. El coste del computador fue de 1.100€. Teniendo en cuenta que el ciclo de vida de un ordenador de tal potencia es de unos 5 años (bastante más larga de lo normal), se podría asociar como coste del proyecto la parte proporcional equivalente a 16,9 meses, es decir, 309,83 €.

Se ha dispuesto también de una conexión a Internet ADSL de alta velocidad a través de fibra óptica, necesaria para búsqueda de documentación y descarga de herramientas, entre otras. Estas son las características de la conexión:

ONO Teléfono + Internet 12Mb  
Velocidad de acceso: 12Mb de bajada / 1Mb de subida  
Router Cisco WIFI monopuesto

El coste mensual de este recurso es de 30,90 €, y dado que el proyecto se ha extendido 16,9 meses (se pagan 17 meses completos), su coste total ha sido de 525,30 €.

Para la realización de las diferentes tareas se ha necesitado el recurso humano y los dos recursos materiales. Estos son los días de trabajo efectivos por tarea:

Toma de requisitos: 2 días.

Análisis: 15 días.

Diseño: 25 días.

Implementación: 33

Experimentación: 226 días.

Documentación: 63 días.

A continuación se muestra una tabla que resume los costes que ha llevado el proyecto:

<b><i>Recurso</i></b>	<b><i>Coste</i></b>
Analista-Programador	32.110 €
Computador	309,83 €
Conexión a Internet ADSL	525,30 €
TOTAL	32.945,13
IVA (21%)	6.918,48 €
<b>TOTAL</b>	<b>39.863,61 €</b>

**Tabla 16: Tabla de costes del proyecto**

Como se puede apreciar en la tabla, el total de gastos asciende a **39.863,61 €**.

## 10. ENTORNO DE DESARROLLO

Para el desarrollo de la aplicación con la que elaborar los ficheros de entrada a la herramienta de minería de datos para realizar los experimentos, se ha elegido el lenguaje Java, siguiendo el paradigma de programación orientación a objetos o POO. El entorno de desarrollo integrado con el que se ha codificado la aplicación es NetBeans, que sirve como editor, compilador, intérprete y depurador.

En cuanto a la versión con la que se compila la aplicación, se ha optado por la Java SE 6, más concretamente por la JDK (Java Development Kit) 1.6.0\_24. La elección de Java para el desarrollo no ha sido difícil, ya que aparte de ser uno de los lenguajes más utilizados en el entorno laboral, ofrece múltiples ventajas. El código que genera el compilador puede ser ejecutado en cualquier entorno que disponga de una máquina virtual de Java (JVM), es totalmente independiente de la plataforma, lo que le otorga portabilidad y versatilidad. Además, esto supone que no hay que realizar ninguna adaptación en el código para que se ejecute correctamente en Windows, Unix o Mac, lo que supone una gran ganancia en el mantenimiento de aplicaciones.

Otro de los puntos fuertes de Java es que es un software de libre distribución y está bajo la licencia GNU. También resulta un lenguaje cuya curva de aprendizaje no es muy pronunciada, su sintaxis es bastante similar a otros lenguajes orientados a objetos como C o C++. Gracias a esto, se pueden encontrar en la red multitud de bibliotecas y librerías creadas por usuarios o equipos de desarrollo con las que poder crear procesos de manera fiable y rápida. Al encuadrarse dentro del paradigma de orientación a objetos, tienen cabida en Java conceptos tales como abstracción, encapsulación, modularidad, herencia de clases o polimorfismo, entre otros.

Una de las características más singulares de Java es que es un lenguaje compilado, pero a la vez también es interpretado. Como el lenguaje fuente es transformado en una especie de código máquina llamado *bytecode*, se dice que es compilado, pero al poder ejecutar dicho código máquina en cualquier plataforma que disponga de un intérprete en tiempo real, resulta también ser un lenguaje interpretado. También admite ejecuciones multihilo, algo especialmente indicado para aplicaciones en entornos distribuidos, por lo que es posible que mientras un hilo se ocupa de efectuar cálculos, otro puede interactuar con el usuario.

La primera versión de Java se presentó en 1995 y a la cabeza del proyecto de Sun Microsystems (actualmente pertenece a Oracle Corporation) se situaba el brillante científico computacional James Gosling. Él fue el encargado de diseñar e implementar el primer compilador y la máquina virtual de Java original. La última versión estable, desplegada el 18 de Marzo de 2014, es la Java SE 8. Como curiosidad, se puede comentar que inicialmente el lenguaje se denominó *OAK*, pero al ser un nombre ya registrado pasó a llamarse *Green*, hasta que finalmente se renombró a Java, tal como se conoce hoy.



Con respecto a NetBeans, se ha utilizado la versión NetBeans IDE 6.9.1. Al igual que Java, la herramienta NetBeans, junto con la también famosa Eclipse, es uno de los entornos de desarrollo integrados más extendidos en el ámbito laboral relacionado con Java. Es una herramienta para desarrolladores muy intuitiva, robusta y amigable, que cuenta con plugins y módulos integrados, lo que supone un gran ahorro en configuraciones del entorno.

La herramienta NetBeans se caracteriza por tener una interfaz de usuario con ventanas, menús y barras de herramientas, lo que proporciona comodidad y agilidad en los desarrollos, y a la vez minimiza las probabilidades de error, pues todas estas características hacen que el desarrollador se centre en las acciones específicas del código.

NetBeans se ofrece en varios paquetes diferentes. Hay ediciones para C y C++, para PHP, para Java SE (Standard Edition) y para Java EE (Enterprise Edition). Este IDE también ofrece herramientas y editores que pueden ser utilizados para HTML, XML y JavaScript, entre otros. Actualmente se puede encontrar soporte para HTML5. También cuenta con soporte para las principales bases de datos, como Oracle, MySQL, Java DB o PostgreSQL, y gracias a la funcionalidad Explorador de Bases de Datos, permite interactuar completamente con ellas, creando, modificando y/o eliminando tablas desde el IDE. Además es multiplataforma, por lo que puede instalarse en Windows, Unix o Mac sin ningún tipo de problema, ya que la propia herramienta NetBeans está escrita en Java, y lo único necesario es que en ese sistema haya instalada una JRE.

La primera versión del IDE NetBeans fue lanzada en Diciembre del año 2000 por Sun Microsystems, diseñada especialmente para la codificación de aplicaciones Java. Como entonces, sigue siendo un producto de código abierto, gratuito y libre, ofrecido bajo las licencias CDDL y GPLv2. La última versión estable desplegada es la 8.0, lanzada el 18 de Marzo de 2014, curiosamente el mismo día que se lanzó la última versión de Java.

Cabe destacar también la utilización de la máquina virtual Oracle JRockit para la consecución de los experimentos. Concretamente, la versión del producto utilizado es Oracle JRockit Real Time 3.1.2 para Java SE 6. Como ya se ha hablado de esta máquina virtual en el presente documento, sólo mencionar la mayor capacidad de control que se tiene sobre las ejecuciones gracias a la funcionalidad Oracle JRockit Mission Control 3.1.2, pues muestra de manera visual aspectos tales como la memoria utilizada, el estado de la pila y su fragmentación o el uso de la CPU.

## 11. GUIA DEL USUARIO

La aplicación ArffGenerator, desarrollada en este proyecto, cuenta con tres funcionalidades independientes, presentadas en el menú principal de la misma. Cabe reseñar que, para mayor comodidad del usuario, en ningún momento se piden rutas de ficheros. Éstas vienen definidas en un fichero de propiedades, que puede actualizarse en cualquier momento.

```
ArffGenerator 1.0  (Generador de ficheros arff para Brown Corpus)
Copyright 2014 Universidad Carlos III de Madrid. Todos los derechos reservados.

Seleccione una de las siguientes opciones:
  1 - Crear diccionario
  2 - Fase 1 de Experimentación
  3 - Fase 2 de Experimentación

Opción: |
```

Figura 35: Menú principal de la aplicación

En la opción número 1, se creará un fichero con todas las palabras distintas de un archivo o archivos (en caso de introducir la ruta de un directorio en el fichero de propiedades) de texto y ordenadas alfabéticamente. La aplicación primero comprobará la existencia de archivos en la ruta especificada, avisando en caso de error. Al finalizar el proceso, dejará el fichero de salida en la ruta descrita en el fichero de propiedades.

```
ArffGenerator 1.0  (Generador de ficheros arff para Brown Corpus)
Copyright 2014 Universidad Carlos III de Madrid. Todos los derechos reservados.

Seleccione una de las siguientes opciones:
  1 - Crear diccionario
  2 - Fase 1 de Experimentación
  3 - Fase 2 de Experimentación

Opción: 1
Comprobando rutas, espere por favor...Correcto!
Creando diccionario, espere por favor...

El proceso de creación del diccionario ha finalizado satisfactoriamente.
Presione la tecla Enter para salir de la aplicación...
```

Figura 36: Pantalla de la opción 1, creación del diccionario

Para crear un fichero con extensión “.arff” para la primera fase de experimentación, el usuario debe seleccionar la segunda opción, tecleando el número “2”. Si se quiere generar sobre el diccionario creado en la primera opción, el usuario, previamente, deberá

trasladar el fichero resultante de dicho punto a la ruta que tenga asignada en el fichero de propiedades. Tras la comprobación de rutas de los archivos necesarios para esta funcionalidad, comenzará el proceso que producirá como salida un fichero de entrada a la herramienta Weka para la primera fase de experimentación.

```
ArffGenerator 1.0    (Generador de ficheros arff para Brown Corpus)
Copyright 2014 Universidad Carlos III de Madrid. Todos los derechos reservados.

Seleccione una de las siguientes opciones:
  1 - Crear diccionario
  2 - Fase 1 de Experimentación
  3 - Fase 2 de Experimentación

Opción: 2
Comprobando rutas, espere por favor...Correcto!
Generando fichero para Fase 1 de Experimentación, espere por favor...

El proceso de creación del fichero para Fase 1 de Experimentación ha finalizado satisfactoriamente.
Presione la tecla Enter para salir de la aplicación...
```

**Figura 37: Pantalla de la opción 2, generación del fichero para Fase 1 de Experimentación**

Como última funcionalidad, el usuario eligiendo la tercera opción y escribiendo el número “3” podrá crear un fichero con extensión “.arff” que sirve para la segunda fase de experimentación. Si el usuario desea crear el fichero de la segunda fase de experimentación con el obtenido en el segundo punto, debe transportar el fichero resultante de la segunda funcionalidad a la ruta especificada en el fichero de propiedades. Una vez comprobadas las rutas de los ficheros necesarios para este apartado, se pedirá al usuario si desea o no incluir los números de las reglas de desambiguación del contexto y que aparezcan en el fichero de salida. Tras teclear “S” (con número de las reglas, se obtendrá un fichero de salida con relaciones de 7 atributos) o “N” (sin el número de las reglas de desambiguación, el fichero de salida cuenta con relaciones de 5 atributos), comenzará el proceso para la generación del fichero de entrada a la herramienta Weka para la segunda fase de experimentación.

```
ArffGenerator 1.0    (Generador de ficheros arff para Brown Corpus)
Copyright 2014 Universidad Carlos III de Madrid. Todos los derechos reservados.

Seleccione una de las siguientes opciones:
  1 - Crear diccionario
  2 - Fase 1 de Experimentación
  3 - Fase 2 de Experimentación

Opción: 3
Comprobando rutas, espere por favor...Correcto!
¿Desea incluir los números de regla de desambiguación para el contexto como atributos?
Opción (S/N): S
Generando fichero para Fase 2 de Experimentación, espere por favor...

El proceso de creación del fichero para Fase 2 de Experimentación ha finalizado satisfactoriamente.
El fichero se ha creado con toda la información procedente del contexto.
Presione la tecla Enter para salir de la aplicación...
```

**Figura 38: Pantalla de la opción 3, generación del fichero para Fase 2 de Experimentación**

## 12. CONCLUSIONES

Tras la realización de este PFC se han obtenido las siguientes conclusiones:

- El grado de acierto de los experimentos se ha visto condicionado por las muestras de datos, pero se puede hablar de éxito en la etiquetación y desambiguación morfológica con reducida información contextual teniendo en cuenta que se ha llegado a superar ampliamente en algunos casos el 94% de acierto. Cabe destacar frente a otras metodologías, que ésta no necesita la aportación de las categorías candidatas de cada palabra, por lo que es muy apropiada para la etiquetación de términos nuevos no presentes en diccionarios. Estos términos son muy comunes en áreas en expansión como la informática, en los que se utiliza gran cantidad de palabras técnicas. Además, trabaja sin o con limitado contexto.
- En el corpus con el que se trabajó en este proyecto, Brown Corpus, se han tenido que separar palabras compuestas para que la desambiguación fuera más efectiva.
- Cuanto mayor sea la muestra de datos, es decir, un corpus de gran tamaño, mejores son los resultados obtenidos. Todo esto debe estar unido a un diccionario de datos lo más completo y variable posible, sobre todo en terminaciones de palabras.
- Para elevar el porcentaje de éxito, se deben eliminar de la muestra, las palabras de otros idiomas. Los signos de puntuación no aportan interés a la investigación, ya que su desambiguación es trivial.
- Como norma general, se ha experimentado una mejoría en los resultados en la segunda parte de experimentación frente a la primera, acentuándose más cuanto mayor es la información que se dispone del contexto de la palabra.
- Los experimentos realizados sobre los extranjerismos encontrados en Brown Corpus han arrojado unos valores de éxito bajos, que incluso iban decreciendo en la segunda fase y más aún cuando se añadía más información del contexto. Esto puede deberse a que la muestra obtenida del corpus era demasiado heterogénea, con términos de diferentes idiomas y a veces sin poder contar con el contexto al ser palabras aisladas.
- El proceso de aprendizaje que se lleva a cabo en la herramienta de minería de datos es de alta complejidad algorítmica y, sobre todo computacional, pues requiere gran cantidad de recursos, tanto en tiempo como en potencia de material. Sin embargo, en producción, los tiempos son muy buenos ya que los modelos construidos son muy completos y resuelven el etiquetado de cada palabra con unas pocas comparaciones.
- Los algoritmos que mejor porcentaje han ofrecido son los meta-algoritmos de Boosting y Bagging, ejecutados sobre algoritmos de reglas (Part) o de árboles de decisión (J48). Posiblemente se hubiera obtenido mayor éxito en los experimentos iniciales si se hubieran realizado con dichos meta-algoritmos. Como generador de reglas de desambiguación, se tomó el resultado del experimento realizado con el meta-algoritmo AdaBoostM1 basado en el algoritmo Part, porque viene expresado en reglas. El resultado del experimento realizado con AdaBoostM1 con el algoritmo J48 era el mejor de la primera fase, y es probable que se hubieran mejorado los resultados en la segunda fase si éste se hubiera elegido para crear las reglas de desambiguación.

## 13. TRABAJOS FUTUROS

Se podrían hacer experimentos utilizando otro corpus en inglés, de manera que se aprendiera con uno y se probase con el otro, y viceversa. O también sobre la unión de ambos corpus.

Dado que se ha elegido un meta-algoritmo basado en el Part para crear las reglas de desambiguación, se podría efectuar el mismo proceso pero eligiendo el mismo meta-algoritmo (AdaBoostM1) basado en J48, que además era el que obtenía mayor porcentaje de aciertos en la primera fase. La complicación reside en que los resultados están en forma de árbol de decisión, por lo que se vuelve más complicado el trabajar con ello.

Se puede realizar un trabajo de desambiguación sobre un corpus en euskera, que es una lengua aislada que no descende de ninguna otra, para comprobar que porcentaje de éxito arroja.

Un trabajo, aunque excesivamente tedioso, sería comprobar manualmente donde están los errores de desambiguación en los resultados de los experimentos realizados en este PFC.

Otro posible trabajo que se puede abordar sería aplicar los resultados obtenidos en este proyecto a una herramienta de los explotara, por ejemplo para obtener un traductor automatizado o un elaborador de resúmenes de texto.

Podría, también, realizarse un pequeño desarrollo para simplificar la redundancia de datos en las reglas de desambiguación obtenidas en la primera fase, de manera que se incrementaría la eficiencia del proceso de creación del fichero para la segunda fase, en el que intervienen dichas reglas, al evitarse cálculos con condiciones repetitivas e innecesarias.

## 14. ANEXOS

### 14.1 Guía rápida para utilización de la herramienta Weka

El único requisito previo que se exige para la utilización de Weka es tener instalado una máquina virtual de Java en el entorno donde se vaya a realizar la experimentación. Tras completar la instalación de la herramienta, se debe abrir la consola de comandos de Windows y navegar hasta el directorio raíz de la aplicación, para a continuación escribir el comando de inicio de la aplicación como aparece en la siguiente figura:

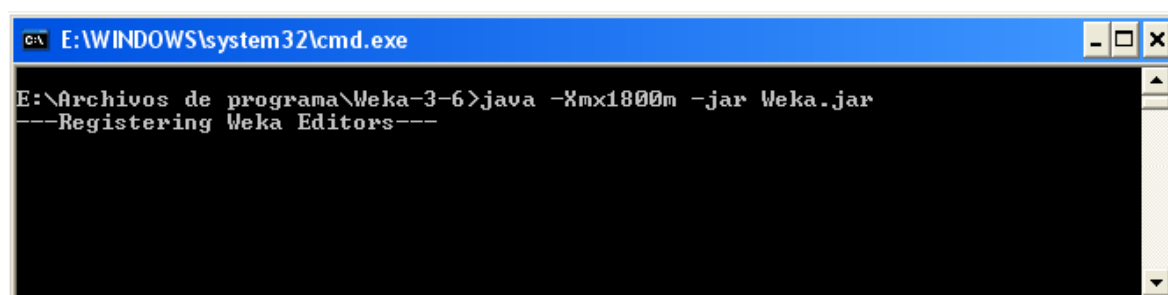
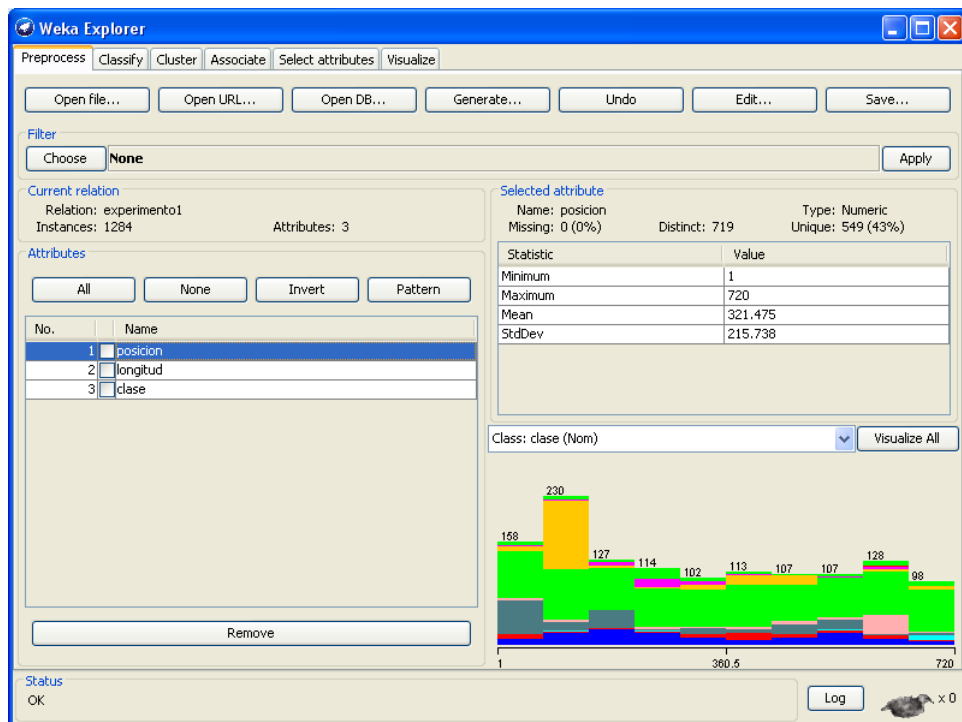


Figura 39: Comando de llamada a Weka

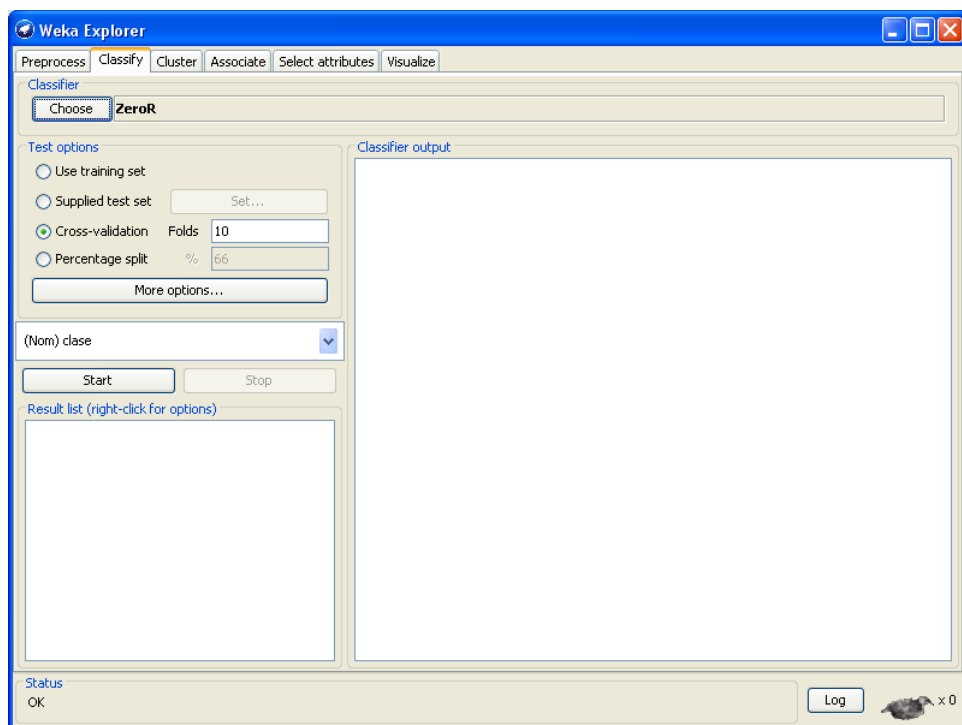
Destacar que en la figura 39 la llamada se hace con la máquina virtual de Java original, mientras que si se decide utilizar la máquina de JRockit habría que sustituir la palabra “java” por la ruta completa del fichero “java.exe” de la instalación de JRockit, aunque también sería conveniente añadirla como variable de entorno. También es reseñable el parámetro “-Xmx”, porque es el encargado de especificar a la máquina virtual el máximo tamaño de memoria principal que se le ha asignado.

Una vez se ha lanzado Weka, se muestra pequeña ventana con el menú de bienvenida. Para realizar los experimentos se debe pulsar en la opción “Explorer” y se abrirá otra ventana de Java con la aplicación Weka Explorer posicionada en la pestaña “Preprocess”. Es en esta pantalla donde se debe introducir el fichero de entrada a la aplicación, es decir, el que contiene los datos con los que se desea experimentar. Para ello, se debe pulsar en el botón “Open file...”, esperar a que se abra el dialogo de Windows para llegar a la ruta de acceso al fichero que se quiere tratar y pulsar en “Abrir”. Si la carga de datos ha sido satisfactoria se mostrará el literal “OK” en la barra de estado de Weka, en la parte inferior de la pantalla. También se mostrará el nombre de la relación, las instancias introducidas y los atributos que las componen. Se mostrará en la parte inferior derecha una gráfica con la distribución de los datos proporcionados en el fichero. Por otro lado, en el cuadro exclusivo de los atributos, se podrá pinchar en cada uno de ellos para poder ver, en la parte derecha, los datos referentes a dicho atributo. Además, desde esta misma pantalla se pueden aplicar filtros a la muestra, pero en este caso no es necesario realizar ningún filtrado con los datos.



**Figura 40: Pantalla de preprocesamiento de Weka**

A continuación se debe pulsar en la pestaña “Classify”, para llegar a la pantalla de selección de algoritmos de clasificación. En este caso se ha elegido el algoritmo ZeroR.



**Figura 41: Pantalla de clasificación de Weka**

Antes de comenzar con la clasificación, se debe elegir la opción “Cross-validation” con 10 pliegues. Seguidamente, se pulsará en el botón “Start” para comenzar. Cuando el

experimento finaliza, se muestra el resultado en la ventana “Classifier output”. He aquí un ejemplo de resultado para el algoritmo “ZeroR”:

```

=== Run information ===

Scheme:weka.classifiers.rules.ZeroR
Relation:  experimento1
Instances:  1284
Attributes:  3
    posicion
    longitud
    clase
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

ZeroR predicts class value: noun

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      642      50  %
Incorrectly Classified Instances    642      50  %
Kappa statistic                    0
Mean absolute error                 0.1414
Root mean squared error             0.2656
Relative absolute error             100  %
Root relative squared error         100  %
Total Number of Instances          1284

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
0      0      0      0      0      0.489  adjective
0      0      0      0      0      0.474  adverb
0      0      0      0      0      0.433  aux
0      0      0      0      0      0.495  determiner
0      0      0      0      0      0.476  invariant
1      1      0.5    1      0.667    0.498  noun
0      0      0      0      0      0.494  preposition
0      0      0      0      0      0.462  pronoun
0      0      0      0      0      0.05   qualifier
0      0      0      0      0      0.491  verb
Weighted Avg.  0.5    0.5    0.25   0.5    0.333    0.492

=== Confusion Matrix ===

 a  b  c  d  e  f  g  h  i  j  <-- classified as
0  0  0  0  0 134  0  0  0  0 | a = adjective
0  0  0  0  0 43  0  0  0  0 | b = adverb
0  0  0  0  0 12  0  0  0  0 | c = aux
0  0  0  0  0 141  0  0  0  0 | d = determiner
0  0  0  0  0 55  0  0  0  0 | e = invariant
0  0  0  0  0 642  0  0  0  0 | f = noun
0  0  0  0  0 162  0  0  0  0 | g = preposition
0  0  0  0  0 35  0  0  0  0 | h = pronoun
0  0  0  0  0 1  0  0  0  0 | i = qualifier
0  0  0  0  0 59  0  0  0  0 | j = verb

```

Los resultados se pueden dividir en cinco partes: la información de ejecución, el modelo generado, resumen de porcentajes, ocurrencias detalladas y matriz de confusión.



## GLOSARIO

**Análisis Morfológico:** Consiste en determinar la forma, clase o categoría gramatical de cada palabra de una oración. No confundir ni mezclar con el análisis sintáctico en el que se determinan las funciones de las palabras o grupos de palabras dentro de la oración.<sup>5</sup>

**Contexto:** El contexto lingüístico se refiere a todos los factores concomitantes (o, que van frecuentemente acompañados) con la producción de enunciados lingüísticos, que afectan a la interpretación, adecuación e incluso significado de dichos mensajes.<sup>6</sup>

**Corpus:** Un corpus o **corpus lingüístico** es un conjunto, habitualmente muy amplio, de ejemplos reales de uso de una lengua. Estos ejemplos pueden ser textos (lo más común) o muestras orales (generalmente transcritas).<sup>7</sup>

**Desambiguar:** Efectuar las operaciones necesarias para que una palabra, frase o texto pierdan su ambigüedad.<sup>8</sup>

**Etiquetar:** Proceso en el que se asigna a una palabra su categoría gramatical.

**Gramática de restricciones:** (en inglés "*Constraint Grammar*", *CG*) es un tipo de gramática que se usa para la desambiguación léxica. También se usa en análisis superficial de oraciones. Normalmente las gramáticas de restricciones tienen más de mil reglas lingüísticas. El concepto fue lanzado por Fred Karlsson en el año 1990, y ahora hay implementaciones para un gran número de idiomas.<sup>9</sup>

**Lingüística computacional:** Campo multidisciplinar de la lingüística y la informática que utiliza la informática para estudiar y tratar el lenguaje humano. Para lograrlo, intenta modelar de forma lógica el lenguaje natural desde un punto de vista computacional. Dicho modelado no se centra en ninguna de las áreas de la lingüística en particular, sino que es un campo interdisciplinar, en el que participan lingüistas, informáticos especializados en inteligencia artificial, psicólogos cognoscitivos y expertos en lógica, entre otros.<sup>10</sup>

**Máquina Virtual de Java:** Una máquina virtual Java (en inglés *Java Virtual Machine*, *JVM*) es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma

---

<sup>5</sup> <http://roble.pntic.mec.es/msanto1/lengua/1anamorf.htm> 20/04/2014

<sup>6</sup> [http://es.wikipedia.org/wiki/Contexto\\_ling%C3%BC%C3%ADstico](http://es.wikipedia.org/wiki/Contexto_ling%C3%BC%C3%ADstico) 20/04/2014

<sup>7</sup> [http://es.wikipedia.org/wiki/Corpus\\_ling%C3%BC%C3%ADstico](http://es.wikipedia.org/wiki/Corpus_ling%C3%BC%C3%ADstico) 20/04/2014

<sup>8</sup> <http://lema.rae.es/drae/?val=desambiguar> 20/04/2014

<sup>9</sup> [http://es.wikipedia.org/wiki/Gram%C3%A1tica\\_de\\_restricciones](http://es.wikipedia.org/wiki/Gram%C3%A1tica_de_restricciones) 20/04/2014

<sup>10</sup> [http://es.wikipedia.org/wiki/Ling%C3%BC%C3%ADstica\\_computacional](http://es.wikipedia.org/wiki/Ling%C3%BC%C3%ADstica_computacional) 20/04/2014

específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java.<sup>11</sup>

**Markov, Modelo Oculto de:** o **HMM** (por sus siglas del inglés, Hidden Markov Model) es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos (u ocultos, de ahí el nombre) de dicha cadena a partir de los parámetros observables. Los parámetros extraídos se pueden emplear para llevar a cabo sucesivos análisis, por ejemplo en aplicaciones de reconocimiento de patrones. Un HMM se puede considerar como la red bayesiana dinámica más simple.

En un modelo de Markov normal, el estado es visible directamente para el observador, por lo que las probabilidades de transición entre estados son los únicos parámetros. En un modelo oculto de Markov, el estado no es visible directamente, sino que sólo lo son las variables influidas por el estado. Cada estado tiene una distribución de probabilidad sobre los posibles símbolos de salida. Consecuentemente, la secuencia de símbolos generada por un HMM proporciona cierta información acerca de la secuencia de estados.

La probabilidad de observar la secuencia  $Y = y(0), y(1), \dots, y(L-1)$  de longitud  $L$  está dada por

$$P(Y) = \sum_X P(Y | X)P(X),$$

donde la sumatoria se extiende sobre todas las secuencias de nodos ocultos  $X = x(0), x(1), \dots, x(L-1)$ . El cálculo por fuerza bruta de  $P(Y)$  es impráctico para la mayoría de los problemas reales, dado que el número de secuencias de nodos ocultos será extremadamente alto en tal caso. Sin embargo, el cálculo puede acelerarse notoriamente usando un algoritmo conocido como el procedimiento de avance-retroceso.

Una notación habitual de un MOM es la representación como una tupla  $(Q, V, \pi, A, B)$ :

- El conjunto de estados  $Q = \{1, 2, \dots, N\}$ . El estado inicial se denota como  $q_t$ . En el caso de la etiquetación categorial, cada valor de  $t$  hace referencia a la posición de la palabra en la oración.
- El conjunto  $V$  de posibles valores  $\{v_1, v_2, \dots, v_M\}$  observables en cada estado.  $M$  es el número de palabras posibles y cada  $v_k$  hace referencia a una palabra diferente.

<sup>11</sup> [http://es.wikipedia.org/wiki/M%C3%A1quina\\_virtual\\_Java](http://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java) 20/04/2014

- Las probabilidades iniciales  $\pi = \{\pi_i\}$ , donde  $\pi_i$  es la probabilidad de que el primer estado sea el estado  $Q_i$ .
- El conjunto de probabilidades  $A = \{a_{ij}\}$  de transiciones entre estados.
  - $a_{ij} = P(q_t = j | q_{t-1} = i)$ , es decir,  $a_{ij}$  es la probabilidad de estar en el estado  $j$  en el instante  $t$  si en el instante anterior  $t - 1$  se estaba en el estado  $i$ .
- El conjunto de probabilidades  $B = \{b_j(v_k)\}$  de las observaciones.
  - $b_j(v_k) = P(o_t = v_k | q_t = j)$ , es decir, la probabilidad de observar  $v_k$  cuando se está en el estado  $j$  en el instante  $t$ .

La secuencia de observables se denota como un conjunto  $O = (o_1, o_2, \dots, o_T)$ .<sup>12</sup>

**Meta-algoritmo:** Una metaheurística es un método heurístico para resolver un tipo de problema computacional general, usando los parámetros dados por el usuario sobre unos procedimientos genéricos y abstractos de una manera que se espera eficiente. Normalmente, estos procedimientos son heurísticos. El nombre combina el prefijo griego "meta" ("más allá", aquí con el sentido de "nivel superior") y "heurístico" (de εὑρίσκειν, heuriskein, "encontrar").<sup>13</sup>

**N-grama:** subsecuencia de  $n$  elementos de una secuencia dada. En el estudio del lenguaje natural se pueden construir los  $n$ -gramas en base a distintos tipos de elementos como por ejemplo fonemas, sílabas, letras, palabras. Los  $n$ -gramas de orden 1 se conocen como unigramas, los de orden 2 pueden llamarse bigramas o digramas y los de orden 3 se denominan trigramas.<sup>14</sup>

**Tagger:** Un etiquetador gramatical (en inglés *tagger*) es una aplicación que se encarga de asignar una etiqueta gramatical a cada una de las palabras de un texto según su categoría léxica.<sup>15</sup>

**Transductor (de estados finitos):** o transductor finito, es un autómata finito (o máquina de estados finitos) con dos cintas, una de entrada y otra de salida. Típicamente las dos cintas de un transductor se ven como una cinta de entrada y otra de salida. Desde este punto de vista, un transductor se dice que transduce (traduce) el contenido de la cinta de entrada a la cinta de salida, mediante la aceptación de una cadena en la cinta de entrada, y

<sup>12</sup> [http://es.wikipedia.org/wiki/Modelo\\_oculto\\_de\\_M%C3%A1rkov](http://es.wikipedia.org/wiki/Modelo_oculto_de_M%C3%A1rkov) 20/04/2014

<sup>13</sup> <http://es.wikipedia.org/wiki/Metaheur%C3%A1stica> 20/04/2014

<sup>14</sup> <http://es.wikipedia.org/wiki/N-grama> 20/04/2014

<sup>15</sup> <http://www.dc.uba.ar/inv/tesis/licenciatura/2013/rodriguez.pdf> 20/04/2014

la generación de otra cadena en la cinta de salida. Esta transducción se puede realizar de forma no determinista y entonces se producirá más de una salida por cada cadena de entrada. En general, un transductor establece una relación entre dos lenguajes formales.<sup>16</sup>

**Viterbi, Algoritmo de:** El algoritmo de Viterbi permite encontrar las secuencias de estados más probable en un Modelo oculto de Markov (MOM),  $S = (q_1, q_2, \dots, q_T)$ , a partir de una observación  $O = (o_1, o_2, \dots, o_T)$ , es decir, obtiene la secuencia óptima que mejor explica la secuencia de observaciones.

Consideremos la variable  $\delta_t(i)$  que se define como:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = i, o_1, o_2, \dots, o_t | \mu)$$

$\delta_t(i)$  es la probabilidad del mejor camino hasta el estado  $i$  habiendo visto las  $t$  primeras observaciones. Esta función se calcula para todos los estados e instantes de tiempo.

$$\delta_{t+1}(j) = \left[ \max_{1 \leq i \leq N} \delta_t(i)(a_{ij}) \right] b_j(o_{t+1})$$

Puesto que el objetivo es obtener la secuencia de estados más probable, será necesario almacenar el argumento que hace máxima la ecuación anterior en cada instante de tiempo  $t$  y para cada estado  $j$  y para ello utilizamos la variable  $\varphi_t(j)$ .<sup>17</sup>

---

<sup>16</sup> [http://es.wikipedia.org/wiki/Transductor\\_de\\_estados\\_finitos](http://es.wikipedia.org/wiki/Transductor_de_estados_finitos) 20/04/2014

<sup>17</sup> [http://es.wikipedia.org/wiki/Algoritmo\\_de\\_Viterbi](http://es.wikipedia.org/wiki/Algoritmo_de_Viterbi) 20/04/2014

## REFERENCIAS

- Aston, G., y L. Burnard. 1997. *The BNC Handbook. Exploring the British National Corpus with SARA*. Edinburgh: Edinburgh University Press.
- Atkins, S., J. Clear y N. Ostler. 1992. "Corpus Design Criteria". *Literary and Linguistic Computing* 7: 1-16.
- Biber, D. 1990. "Methodological Issues Regarding Corpus-bases Analyses of Linguistic Variation". *Literary and Linguistic Computing* 5/4: 257-269.
- Biber, D. 1993. "Representativeness in Corpus Design". *Literary and Linguistic Computing* 8: 243-57.
- Biber, D., S. Conrad y R. Reppen. 1998. *Corpus Linguistic. Investigating Language Structure and Use*. Cambridge: Cambridge University Press.
- Bunt, H., y M. Tomita, eds. 1996. *Recent Advances in Parsing Technology*. Dordrecht: Kluwer.
- Busa, R. 1974-1980. *Index Thomisticus*. Stuttgart: Fromman-Holzboog. [en CD-ROM desde 1992]
- Charleston, B.M. 1960. *Studies on the emotional and affective means of expression in modern English*. Bern: Francke Verlag.
- Chomsky, N. 1965. *Aspects of the Theory of Syntax*. Cambridge, Mass.: The MIT Press.
- Close, R.A. 1975. *A reference grammar for students of English*. London: Longman.
- Dermatas y K. George. 1995. *Automatic stochastic tagging of natural language texts*. *Computational Linguistics*, 21(2):137-163
- Ekbal, Asif, y S. Bandyopadhyay. 2007. *Lexicon Development and POS tagging using a Tagged Bengali News Corpus*, In *Proc. of FLAIRS-2007*, Florida, 261-263
- Ekbal, Asif, Haque, R. y S. Bandyopadhyay. 2008. *Named Entity Recognition in Bengali: A Conditional Random Field Approach*, In *Proc. of 3rd IJCNLP*, 51-55
- Ekbal, A. Bandyopadhyay, S. 2008. *Part of Speech Tagging in Bengali Using Support Vector Machine*, ICIT- 08, IEEE International Conference on Information Technology, pp. 106-111
- Francis, W.N. y H. Kucera. 1961. *A standard corpus of present-day edited American English*. Providence, Rhode Island: Brown University.
- Gurpreet Singh. 2008 . *Development of Punjabi Grammar Checker*. Phd. Dissertation.

Haegeman, L. 1986. "The present subjunctive in contemporary British English", *Studia Anglica Posnaniensia*, 19, 61-74.

Harsh, W. 1968. *The subjunctive in English*. Alabama: University of Alabama Press.

James Allen. 1995. *Natural Language Understanding*. Benjamin/Cummings Publishing Company.

Jelinek, F. 1985. *Markov source modeling of text generation*. In J.K. Skwirzynski (ed.), *The Impact of Processing Techniques on Communications*, E91 of NATO ASI series.

Johansson, S. 1991. *Computer Corpora in English Language Research*. Eds. Johansson y Stenström. 3-13

Johansson, S., y A-B. Stenström, eds. 1991. *English Computer Corpora. Selected Papers and Research Guide*. Berlin: Mouton.

Jurafsky D and Marting J H. 2002. *Speech and Language Processing An Introduction to Natural Language Processing*. Computational Linguistics and Speech Recognition, Pearson Education Series.

Kennedy, G. 1998. *An Introduction to Corpus Linguistics*. London: Longman.

Kretzschmar, W., Ch. Meyer y D. Ingegneri. 1997. "Uses of Inferential Statistics in Corpus Studies". Ed. Ljung. 167-78.

Lawler, J y H. Dry. 1998. *Using Computer in Linguistics. A Practical Guide*. London: Routledge.

Leech, G., T. McEnery y M. Wynne. 1997. "Further Levels of Annotation". Eds. Garside et al. 85-101.

Leech, G y R. Garside. 1991. "Running a Grammar Factory. The Production of Syntactically Analysed Corpora or 'Treebanks'". Eds. Johansson y Stenström. 15-32.

Ljung, M., ed. 1997. *Corpus-based Studies in English. Papers from the Seventeenth International Conference on English Language Research on Computerized Corpora (ICAME 17), Stockholm, May 15-19, 1996*. Amsterdam: Rodopi.

McEnery, T., y A. Wilson. 1996. *Corpus Linguistics*. Edinburgh: Edinburgh University Press

Pérez, Ch., A. Moreno y P. Faber. 1995. "Lexicografía computacional y lexicografía de corpus".

Qiao, H.L. 1995. "The Mapping between the Parsing Annotation Schemes of the Lancaster Parsed Corpus and the Susanne Corpus". *ICAME Journal* 19: 63-91

Quirk, R., S. Greenbaum, G. Leech & J. Svartvik. 1972. *A grammar of contemporary English*. London, Longman.

Quirk, R. y J. Svartvik. 1980. *A corpus of English conversation*. Lund Studies in English. Lund: CWK Gleerup

Sampson, G. 1995. *English for the Computer. The SUSANNE Corpus and Analytic Scheme*. Oxford: Clarendon Press.

Sánchez, A y P. Cantos. 1997. "Predictability of Word Forms (Types) and Lemmas in Linguistic Corpora. A Case Study Based on the Analysis of the CUMBR Corpus: An 8-million-word Corpus of Contemporary Spanish". *International Journal of Corpus Linguistics* 2/2: 259-80.

Souter, C., y T.F. O'Donoghue. 1991. "Probabilistic Parsing in the COMMUNAL Project". Eds. Johansson y Stenström. 33-48.

Strang, B. 1968. *Modern English Structure*, 2nd ed., revised (1st ed., 1962). London: Edward Arnold.

Stubbs, M. 1996. *Text and Corpus Analysis*. Computer-assited Studies of Language and Culture. Oxford: Blackwell.

Wilson, A., y J. Thomas. 1997. "Semantic Annotation". Eds. Garside *et al.* 53-65.

Wisniewski, E.J. & G.L. Murphy. 1989. *Superordinate and basic category names in discourse: A textual analysis*. Discourse Processes 12: 245-261.

Yates, A.R. 1977. Text compression in the Brown Corpus using variety-generated keysets, with a review of the literature on computers in Shakespearean studies. M.A. dissertation, University of Sheffield.

Zettersten, A. 1968. *Current computing activity in Scandinavia relating to language and literature research*. Computers and the Humanities 3: 53-60.

Zettersten, A. 1969. *A statistical study of the graphic system of present-day American English*. Lund: Studentlitteratur.